Desenvolvimento de uma Solução Robótica para Operações de Acabamento de Solas de Sapatos

Diogo Gigante Viana

Dissertação do MIEM

Orientadores:

Prof. Manuel Rodrigues Quintas

Prof. Paulo Augusto Ferreira de Abreu



Faculdade de Engenharia da Universidade do Porto

Mestrado Integrado em Engenharia Mecânica

Opção de Automação

Julho de 2010

Aos meus amigos

Resumo

Nos tempos que correm, apesar da forte concorrência estrangeira, o calçado Português continua a ter uma elevada presença internacional, sendo um dos sectores que mais contribui para a balança comercial do país. Contudo a organização empresarial deste sector assenta em pequenas e médias empresas com uma forte componente de trabalho manual. Esta estrutura constitui fortes entraves à automatização dos processos tecnológicos, que condicionam a produtividade, conferindo-lhe uma forte dependência de mão-de-obra qualificada e reduzida flexibilidade.

Factores como estes motivaram este projecto onde foi desenvolvido um dispositivo de custo reduzido, passível de ser integrado numa célula robotizada, para acabamento de sapatos.

Após uma introdução de contextualização deste projecto, é apresentada, no segundo capítulo, uma pesquisa sobre os sistemas robotizados comerciais e experimentais actualmente aplicados nas diversas etapas de produção industrial de calçado.

Seguidamente, no terceiro capítulo, é elaborado um estudo de soluções da configuração da célula robotizada, e do funcionamento do dispositivo de acabamento de solas de sapatos, atendendo aos requisitos propostos.

Após este estudo, é apresentado, no quarto capítulo, o protótipo do dispositivo de acabamento desenvolvido. Este dispositivo possui um grau de liberdade, o qual transporta uma ferramenta de corte. Possui ainda um sistema pneumático de regulação de força, um sistema electrónico de medição do deslocamento e uma comunicação de dados através de uma porta série.

Visando a validação do dispositivo foram realizadas simulações dos processos de acabamento com recurso à programação *offline* do *software* de simulação interactiva da *ABB*, o *RobotStudio*.

Efectuadas as simulações, o dispositivo foi então integrado numa célula robotizada, com um manipulador *ABB* de seis eixos, onde foram realizados vários ensaios de validação do dispositivo e do sistema de acabamento de solas de sapatos desenvolvido.

Development of a Robotic Solution for Footwear Finishing Operations

Abstract

These days, despite strong foreign competition, the Portuguese footwear continues to have a high international links, and this sector is the one who contributes most to the country's trade market. However, the business organization of this sector is based on small and medium enterprises with an intensive manual labor. This structure has strong barriers to the automation of technological processes, which affect productivity, giving it a strong dependence of labor-skilled personal and low flexibility.

These reasons have led to the execution of this project, where a low cost machining device was developed, with the aim of being integrated in a robotic cell for finishing operations of shoes.

After an introduction of this project, is presented on the second chapter, a research on currently commercial and experimental robotic systems, applied at various stages of manufacturing processes of footwear.

This is followed by a study of the solutions for the operation of the finishing device, given the proposed requirements.

After this study, was built a prototype of the finishing device, with one degree of freedom under which it carries a cutting tool. The device has also a pneumatic passive force control and an electronic measurement of displacement and data communications system through a serial port.

Once the construction was terminated, the finishing processes were simulated using the offline programming software for interactive simulation from *ABB*, the *RobotStudio*.

Made the simulations, the device was then integrated into a robotic cell with a six-axis *ABB* manipulator, being performed several validation tests of the device and the finishing system developed.

Agradecimentos

Começo por expressar aqui um profundo agradecimento aos meus orientadores, Professor Paulo Abreu e Professor Manuel Quintas, pelo seu apoio, disponibilidade e incansável dedicação, ao longo de todo este semestre. Pelas longas horas que despenderam comigo na procura de soluções para os problemas que foram surgindo com o desenrolar deste projecto, e pela partilha de conhecimentos que me ajudaram em várias fases a melhorar este trabalho, deixo aqui o meu sincero Obrigado.

Agradeço também ao coordenador da opção de Automação, o Professor Francisco Freitas, pelo acompanhamento feito ao longo deste período, e pelas suas críticas construtivas.

Gostaria ainda de agradecer aos meus colegas da opção de Automação, em especial ao Tiago Ramos, ao Filipe Abrantes e ao Nuno Moita, pela ajuda que deram e pelos bons momentos de companheirismo e descontracção.

Aos meus amigos em geral, aos colegas com quem partilhei casa, aos meus amigos de Erasmus e ao *"Pobão"* de Viana, pelo apoio e pelas recordações que temos vindo a construir juntos nestes últimos anos e que jamais esquecerei.

Por fim gostaria de agradecer à minha família, em especial ao meu irmão, aos meus pais, e aos meus avós, que são um exemplo de vida para mim e sempre me apoiaram em todas as circunstâncias, fazendo-me muitas vezes ver que nem sempre o caminho mais fácil é o mais proveitoso.

Índice de Conteúdos

Índice de Figuras

Índice de Tabelas

1 . Introdução Geral e Objectivos

1.1 Introdução

Com uma forte representatividade no sector industrial português, o sector do calçado continua a ser uma das indústrias que mais contribui para a balança comercial do país.

Após uma fase de consolidação, este sector atravessa agora um período de reestruturação, motivada pelas crescentes exigências dos consumidores e pelo aparecimento de novos mercados internacionais que, associados à actual crise económica alteraram as realidades competitivas.

Perante uma incapacidade de competir com outros países nos custos de produção associados, a aposta estratégica em segmentos de mercado mais exigentes e mais valorizados, a par do registo de novas marcas e modelos, fizeram com que o calçado português evoluísse na cadeia de valor, estando já entre os mais valorizados do mundo, diferenciando-o da restante oferta internacional. Como é visível na Figura 1.1, esta aposta na qualidade, aliada à deliberada aposta em determinados mercados externos, faz com que o peso das exportações seja vital para esta indústria, conseguindo assim enfrentar poderosos concorrentes internacionais, nomeadamente os asiáticos.

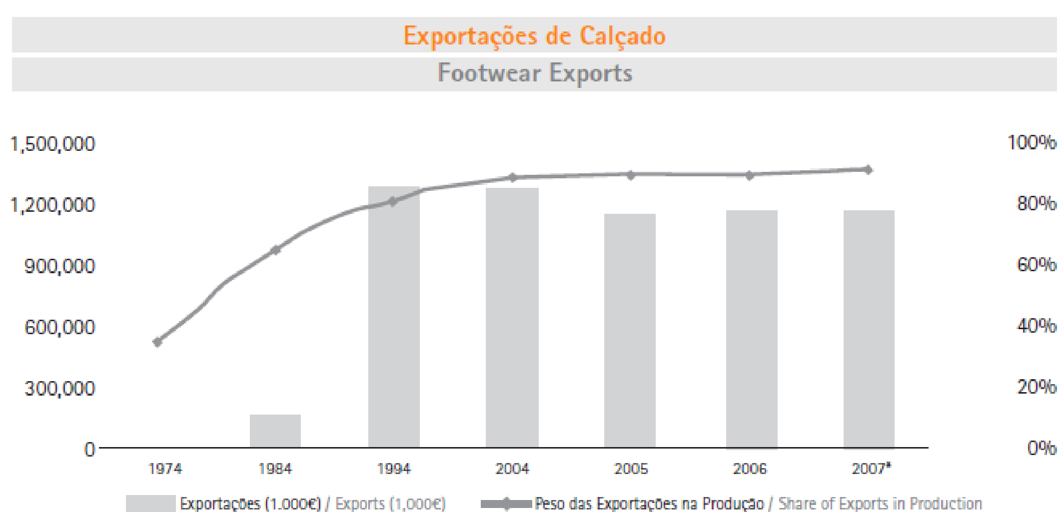


Figura 1.1 – Exportações de calçado em Portugal[1]

[1] Figura retirada de APPICAPS – Monografia estatística'07.

A exportação, que nos primeiros 11 meses de 2009 rondou os 1 167 milhões de euros, é o caminho que as indústrias do sector querem continuar a seguir, tendo para isso também planos para uma maior divulgação junto de mercados emergentes, como Brasil, China, Japão, EUA, Emirados Árabes Unidos, Polónia e Rússia.

A participação de um grande número de empresas em feiras e certames com elevado reconhecimento internacional tem vindo a aumentar consideravelmente de ano para ano, pretendendo os grupos nacionais mostrar "uma indústria jovem, moderna e voltada para o futuro, que alia a tradição às tecnologias de ponta, o 'saber fazer' ao melhor design de sempre", como afirma a Associação Portuguesa dos Industriais de Calçado, Componentes, Artigos de Pele e seus Sucedâneos – APPICAPS. No total, cerca de centena e meia de empresas estão empenhadas na divulgação do seu calçado a nível internacional, associadas a um investimento global na promoção externa do sector, que desde 2008 até 2010 rondou os 24 milhões de euros. Investimento com retorno, se fizermos as contas à contribuição do sector do calçado para a balança comercial portuguesa, que foi o que contribuiu mais positivamente nos primeiros 11 meses de 2009[2].

Contudo, apesar dos indicadores se apresentarem optimistas, a indústria portuguesa do calçado apresenta ainda algumas limitações. É maioritariamente constituída por pequenas empresas, localizadas na região norte do país, com uma forte componente de trabalho manual, requerendo trabalhadores experientes.

A constante introdução de novos modelos de sapatos a cada época do ano, acatando os requisitos transitórios da moda, faz com que seja um sector em constante adaptação, o que acaba por contrastar um pouco com a própria estrutura deste sector em Portugal.

O Processo Produtivo Tradicional

O processo produtivo na indústria de calçados caracteriza-se pela sua descontinuidade, com o fluxo de produção ocorrendo entre estágios bastante distintos entre si. Para compreender melhor o método de produção de um sapato, enumeram-se e explicam-se em seguida sucintamente as cinco principais etapas deste processo, que são: a modelação, o corte, a costura, a montagem e o acabamento.

[2] Jornal de Noticias - edição online de 02-03-2010 (http://jn.sapo.pt/PaginaInicial/Economia/Interior.aspx?content_id=1508198).

Durante a fase de modelação, evidenciada na Figura 1.2, cabe ao estilista/designer idealizar o produto final, considerando aspectos como as tendências da moda, os materiais a serem utilizados, a definição dos modelos e das formas que compõem o calçado, adaptados a factores como o conforto, a aparência, a durabilidade e o estilo. Além disso, tem a função de procurar adequar a manufacturabilidade do produto, adaptando a sua concepção às condições e características do processo produtivo, inclusive no que tange a custos.



Figura 1.2 – Modelação de um sapato[3]

Na etapa de corte (Figura 1.3 A), a matéria-prima é cortada de acordo com as determinações definidas na modelação. No processo tradicional, o corte é realizado com ferramentas adequadas, de uma forma manual. Em processos automatizados existem equipamentos industriais que permitem a execução desta operação de uma forma automática.

Depois do corte das peças, estas são unidas na etapa da costura (Figura 1.3 B). Nesta fase, de acordo com o tipo de calçado, as várias peças que compõem o cabedal são costuradas, dobradas, picotadas ou coladas, podendo também ser aplicados enfeites e fivelas.



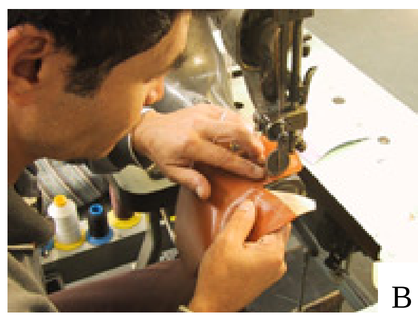


Figura 1.3 – A – Operação de corte[3]; B - operação de costura[3]

[3] Imagens retiradas de http://www.dipollini.com.br/home/inst_comoefeito.asp em 10/01/2010.

As etapas constituintes da montagem podem ser variadas consoante os modelos de sapatos fabricados (Figura 1.4). Na fase de montagem, o cabedal é unido à sola. Os processos de união são bastante variados, envolvendo costura, prensagem e/ou colagem. A colocação de saltos, biqueiras e palmilhas também é realizada nesta fase.



Figura 1.4 – Fase de montagem de um sapato[4]

No acabamento (Figura 1.5), o calçado é desenformado e passa pelos retoques finais, como a colocação do forro, pintura, enceramento e uniformização das superfícies.

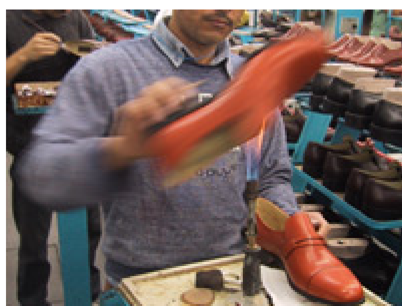


Figura 1.5- Acabamento de um sapato[4]

Em cada uma destas etapas, as operações realizadas também são bastante variadas, de acordo com o tipo de calçado produzido.

Automação do Processo Produtivo

De uma maneira geral, pode-se perceber que, embora a difusão da robótica tenha exercido impactos importantes sobre o processo de fabricação de calçados, esses impactos

---

[4] Imagens retiradas de http://www.dipollini.com.br/home/inst_comoefeito.asp em 10/01/2010.

ocorreram de forma mais intensa em algumas etapas do processo produtivo dependendo do produto em questão[5].

Também a introdução contínua de novos modelos de sapatos no mercado a cada estação do ano, atendendo às inconstantes tendências da moda, constitui uma limitação à automação dos processos tecnológicos de fabrico.

Convém ainda referir que os processos de produção dos calçados de couro e de material sintético tem requisitos distintos. Estes últimos, devido às características da matéria-prima e ao facto de serem produzidos em grandes séries, permitem e justificam a adopção de sistemas de automação para a sua produção. O calçado de couro possui um carácter semi-artesanal o que condiciona a adopção de soluções de automação.

Em etapas, como a modelagem e o corte, é possível utilizar equipamentos como o CAD, inclusive de forma integrada com equipamentos automatizados de corte e manufactura, especialmente quando se trata do processamento de materiais sintéticos ou de couro de qualidade mais elevada, cujos requisitos de uniformidade são bem mais elevados[5].

Porém, outras fases como a costura, a montagem e o acabamento, devido à natureza do produto em questão e também à estrutura empresarial do sector nacional do calçado, estão ainda muito dependentes de processos com carácter artesanal intensivo. Estes processos necessitam de mão-de-obra experiente, focalizada em determinados estádios de concepção do produto, a maior parte das vezes com cadências de produção distintas, que conduzem a uma descontinuidade do processo de fabrico.

Enumeram-se, de seguida, as operações do processo produtivo de sapatos nas quais os sistemas de automação se encontram mais difundidos:

- Corte;
- Pulverização e colagem;
- Desbaste ou cardagem;
- Cosedura da sola.

[5] Coutinho, *"Sector do calçado - Diagnóstico e Propostas para o Incremento da Competitividade Industrial com Base no Design"* (http://www.spdesign.sp.gov.br/principal.htm) visto em 11/12/2009.

Acabamento das solas duplas

Os processos de obtenção e acabamento das solas dos sapatos diferem consoante o tipo de calçado em questão. Podem ser obtidas por processos de corte, moldação indirecta, moldação directamente no sapato, entre outros.

Após a montagem do sapato, a sola superior é cozida à base deste e, posteriormente, é colada a sola inferior à sola superior. Um exemplo de um sapato com sola dupla é exibido na Figura 1.6.



Figura 1.6 – Sapato de vela com sola dupla

Normalmente após a colagem a sola superior apresenta um desfasamento em relação à inferior, rondando o milímetro em todo o contorno do sapato. Geralmente é a sola superior que se salienta em relação à sola inferior.

A operação de acabamento das solas duplas consiste na uniformização das duas superfícies, de maneira a que se assemelhem geometricamente a uma só. Esta operação é realizada manualmente com um operador que encosta e roda a sola do sapato relativamente a uma ferramenta abrasiva, tipicamente uma lixa de fita (Figura 1.7).



Figura 1.7 – Processo de acabamento de um sapato (A - antes; B – processo[6]; C - depois)

[6] Imagem retirada de http://custommadeshoes.com/images/workshop/workshop10.jpg em 10/10/2009.

Após um contacto anterior com uma empresa do ramo sediada em Felgueiras, a *Armipex*, fabricante deste tipo de sapatos de vela com sola dupla para uma conhecida multinacional, foi identificada esta necessidade de automatização de operações de acabamento de solas de sapatos, o que levou ao desenvolvimento deste projecto.

Este trabalho vem no seguimento de um projecto realizado anteriormente com um propósito semelhante, que envolveu a concepção de uma célula robotizada para acabamentos de solas de sapatos, recorrendo a uma solução robótica com um robô controlado em força[7]. Contudo o presente trabalho difere significativamente no que diz respeito à procura da solução de controlo de força a implementar. Assim, neste trabalho pretende-se utilizar um robô standard, controlado em posição/velocidade da trajectória, e não em força, pelo que será necessário o desenvolvimento de um dispositivo auxiliar para o controlo de força.

1.2 Objectivos do trabalho

O objectivo deste trabalho é o desenvolvimento de um sistema automático de acabamento de solas em sapatos, baseado numa solução robótica que responda aos seguintes requisitos:

- Utilização de um robô standard, operado em controlo de posição/velocidade;
- Comando do robô em controlo de posição/velocidade;
- Desenvolvimento de um dispositivo auxiliar de acabamento com capacidade de regulação da força de contacto entre a ferramenta e o sapato.

Assim torna-se necessário o desenvolvimento de um dispositivo auxiliar de acabamento, projectado com um sistema que permita controlar a força de contacto entre a ferramenta e o sapato.

Os testes de simulação e implementação desta solução automatizada foram efectuados recorrendo ao robô industrial *ABB IRB 2400*, disponível no laboratório de robótica do Departamento de Engenharia Mecânica da Faculdade de Engenharia da Universidade do Porto.

[7] Dissertação em Automação intitulada "*Concepção e Simulação de uma Célula Robotizada para Acabamentos de Solas de Sapatos*", realizada por Nuno Moita em 2009, na Faculdade de Engenharia da Universidade do Porto.

1.3 Organização dos temas abordados no relatório

Este relatório está estruturado em seis capítulos, seguidos da Bibliografia e Anexos.

Assim, após este primeiro capítulo introdutório segue-se o segundo capítulo, denominado ***Estado da Arte***, onde se apresenta o resultado de uma pesquisa sobre os sistemas robotizados existentes na Indústria do calçado.

O terceiro capítulo, intitulado ***Arquitectura do Sistema Robótica Para Operações de Acabamento***, mostra o estudo das soluções e configurações idealizadas para uma mais adequada e eficiente concepção do processo de acabamento das solas dos sapatos, atendendo aos requisitos do projecto.

O quarto capítulo, denominado ***Desenvolvimento de um Dispositivo de Controlo de Força***, evidencia todas as etapas do projecto do dispositivo auxiliar que vai ser utilizado para fazer a operação de acabamento das solas dos sapatos.

O quinto capítulo, intitulado ***Implementação do Sistema Desenvolvido***, relata a integração funcional dos componentes desenvolvidos na célula robotizada e a sua programação. Todos os procedimentos efectuados para testar e validar a solução proposta são detalhadamente comentados, bem como os ensaios efectuados, e os respectivos resultados experimentais.

Por fim, no sexto capítulo, são tecidas as ***Conclusões*** finais com base no trabalho desenvolvido e são sugeridos ***Trabalhos Futuros*** que poderão eventualmente dar continuidade a este projecto.

2 . Estado da Arte

2.1 Introdução

A preferência de muitos fabricantes de sapatos ainda recai sobre a utilização de máquinas dedicadas para as diferentes etapas de fabrico, máquinas essas que necessitam de um operador mais ou menos qualificado para as operar.

Contudo, este projecto iniciou-se com uma recolha de informação exaustiva, centrada em sistemas industriais automáticos, disponibilizados pelos fabricantes e utilizados na indústria do calçado. Essa pesquisa funcionou como ponto de partida para este trabalho, possibilitando perceber em que ponto se encontrava o "estado da arte", identificar eventuais lacunas de mercado e eventuais limitações dos sistemas apresentados.

2.2 Sistemas robotizados comerciais

DESMA

A *DESMA* é uma empresa alemã especializada no design de linhas de produção automáticas para a injecção de solas directamente no calçado, fornecendo também outras soluções de automação para determinadas operações que envolvam o fabrico de calçado com sola de borracha. Efectua projectos de linhas de produção totalmente automatizadas, fazendo também a integração de sistemas automáticos em linhas de produção já existentes. É possivelmente a maior fornecedora de soluções robotizadas comerciais para o ramo do calçado.

A *DESMA* apresenta soluções de automação para as diferentes fases de produção aplicadas :

- ao molde – aplicação de agentes desmoldantes e transporte/posicionamento da sola;
- à parte superior do sapato – cardagem, aplicação de cola e transporte/posicionamento da sola;
- à sola – aplicação de cola e manipulação da sola;
- ao sapato – recorte do forro e manipulação do sapato.

A *DESMA* desenvolve as próprias ferramentas e sistemas para aplicação em soluções robotizadas. Essas soluções robotizadas envolvem:

- **a aplicação de agentes desmoldantes**

A utilização de um agente desmoldante é necessária para evitar a adesão do poliuretano injectado no molde, permitindo remover a sola do molde com facilidade e sem danos. Combinado com o agente desmoldante, o robô utiliza um sistema electrostático para realizar esta operação. Esta tecnologia reduz o consumo do agente de desmoldante, aumentando a vida útil do molde.

- **a cardagem**

A cardagem permite uma melhor adesão entre os materiais a serem colados. O material que constitui a parte superior do sapato deve ser trabalhado a velocidades diferentes e com pressões de contacto a variar de acordo com o tipo de ferramenta utilizada. Por este facto, a força exercida pelo robô que realiza a cardagem deve ser então cuidadosamente controlada. A cabeça do robô possui uma ferramenta rotativa cuja força de contacto é controlada segundo a linha de contorno previamente definida.

- **a aplicação de cola**

Após a cardagem, quer a sola seja unida à parte superior por colagem ou por injecção/moldagem, é necessária a utilização de agentes adesivos. Em ambos os casos, o componente adesivo pode ser pulverizado uniformemente e com uma espessura adequada, ou aplicado segundo uma única linha, por exemplo, na extremidade da sola (Figura 2.1).



Figura 2.1– Robô para aplicação de cola[8]

[8] Imagem retirada de “*Trasferimento tecnológico per l'Automazione nel Settore Calzaturiero*”, Distretto Calzaturiero Veneto, Progetto di Ricerca: Relazione Finale 2008.

- **a alimentação / recolha de sapatos**

Naturalmente, esta empresa também oferece robôs que manipulem os produtos. Estes robôs para movimentação e posicionamento podem ser utilizados de diferentes maneiras e em vários estágios de produção, permitindo optimizar o fluxo de material dentro da actividade produtiva.

O *Flex Pendant*

Esta empresa, em parceria com a *ABB*, desenvolve desde 2002 aplicações específicas para serem executadas na consola de programação/operação dos manipuladores *ABB*, o *Flex Pendant* (Figura 2.2).

Este dispositivo tem várias funcionalidades e, utilizando as aplicações desenvolvidas pela *DESMA*, é permitido a um operador não especializado controlar várias estações e modificar, durante a produção, o programa em execução com um simples toque no ecrã, eliminando a necessidade de se reiniciar o controlador.



Figura 2.2 - *Flex Pendant – ABB*

ACTIS

Esta empresa francesa de nome *ACTIS*, centra-se na concepção de linhas de montagem e de fabrico automático de células robotizadas e sistemas de gestão integrada de logística. As células individuais robóticas podem ser utilizadas tanto para controlar sistemas de injecção de plástico, como para implementar algumas fases do trabalho ao longo de uma linha manual.

Todos os meios que levam à criação de uma linha automática (ou uma única célula) são parte de um sistema denominado "*RB System*", que inclui, portanto, todos os

equipamentos, processos, *software* e sistemas robotizados, desenvolvidos especificamente para a automatização do processo de produção na indústria de calçado.

O "*RB System*" não inclui, no entanto, só produtos padronizados, pois cada componente é feito de acordo com necessidades específicas de cada cliente.

A empresa utiliza um sistema de identificação por etiquetas electrónicas e faz uso de um sistema, denominado "*Prehensor*", que permite a manipulação da posição no espaço das formas e o seu transporte, mantendo referências exactas. Além disso, as etiquetas estão também equipadas com *chips* electrónicos que permitem atribuir códigos de identificação para cada sapato, contendo informação variada sobre cada modelo ao longo das diferentes fases de produção.

Desde a sua entrada no campo da automação aplicada ao calçado, a *ACTIS* já registou mais de 80 patentes. De entre os vários sistemas desenvolvidos, dispõe de soluções robóticas para cardar (nos lados e fundo do sapato), para a colagem, para o corte, para o carregamento e descarregamento de máquinas de moldagem por injecção assim como para a libertação do molde e lubrificação automática dos moldes, entre outros.

Os robôs para corte e cardagem possuem uma ferramenta equipada com um sensor de carga, a fim de garantir o contacto permanente e a força constante da ferramenta com os limites do sapato.

Para a realização das células e linhas automatizadas (Figura 2.3), a fim de assegurar a precisão necessária para a produção de calçado, a empresa faz uso extensivo de robôs industriais *Stäubli* de 6 graus de liberdade.



Figura 2.3 – Linha automática de cardagem – *ACTIS*[9]

[9] Imagem retirada de Kochan, [Kochan,1996].

Robot System

Esta empresa italiana, fundada em 1984, é especializada na concepção e construção de linhas automáticas para a produção de calçado com sola em borracha. As linhas automáticas instaladas, denominadas linhas *FMSS* (*Flexible Manufacturing Shoe System*), como o nome indica, são constituídas por um sistema integrado flexível para automatizar o processo de produção. A sua estrutura assenta num conjunto modular e configurável constituído por ilhas associadas a sistemas de transporte (Figura 2.4).



Figura 2.4- Linha de fabrico da *Robot System*[10]

Tal como em outras empresas do mesmo ramo, para além de soluções genéricas para a automação de algumas etapas do processo de fabrico, a *Robot System* desenvolve também soluções à medida das necessidades dos clientes.

Esta empresa constrói as suas próprias ferramentas aplicadas aos manipuladores, e dedica-se também à criação e desenvolvimento de *software* de programação *off-line*, para as suas células robotizadas.

Na realização das linhas automáticas, a *Robot System* utiliza manipuladores Kawasaki de 6 eixos.

[10] Imagem retirada de "*Trasferimento tecnológico per l'Automazione nel Settore Calzaturiero*", Distretto Calzaturiero Veneto, Progetto di Ricerca: Relazione Finale 2008.

AUTEC Automation

A *AUTEC Automation* foi fundada em 1993 com o intuito de fabricar soluções automatizadas para a indústria do calçado. Esta empresa italiana fornece várias soluções de automação para a indústria de calçado, desde sistemas automáticos isolados de cardagem, injecção de solas, desmoldação e extracção das solas dos moldes, sistemas de transporte, identificação por RFID, até ilhas e linhas completas de produção (Figura 2.5).

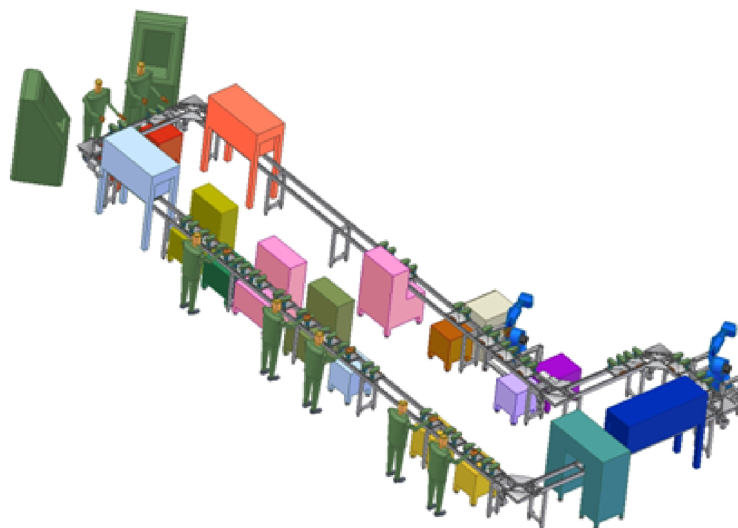


Figura 2.5 – Linha de produção *AUTEC*

A *AUTEC Automation* desenvolve processos automatizados, como por exemplo o de cardagem mostrado na Figura 2.6, com um dispositivo de desbaste equipado com um sistema de compensação da ferramenta, que mantém uma pressão constante e uniforme sobre o cabedal. A definição do ciclo de trabalho específico para cada sapato é efectuada mediante a introdução de apenas dois dados: o número do molde e a referência do sapato.



Figura 2.6 – Operação de cardagem *AUTEC*

Esta empresa desenvolveu também um sistema de visão para integração com um robô que permite a programação automática das trajectórias para uma determinada operação. Desenvolveu ainda um digitalizador capaz de identificar o perfil das solas ou dos sapatos (Figura 2.7). O digitalizador possui um apalpador, em que a sua parte terminal é arrastada manualmente pelo operador. As cotas são automaticamente detectadas e registadas. Isto possibilita o ajustamento das cotas antes de o molde entrar em produção, podendo os dados ser transmitidos a todas as máquinas.

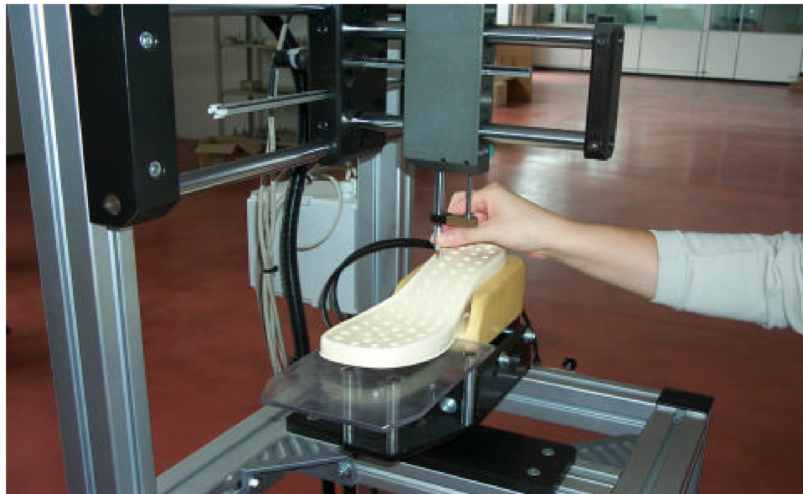


Figura 2.7 – Digitalizador - *AUTEC*

Intelligent Machines Corporation – IMC

Esta empresa Americana, sediada no estado da Florida, foi criada por antigos cientistas e engenheiros do programa espacial da NASA. O seu objectivo, ao contrário das suas concorrentes europeias, é encontrar soluções de automação para as etapas produtivas mais inconstantes e com maiores restrições, na indústria do calçado.

Contrariamente aos sistemas desenvolvidos pela *DESMA* e pela *ACTIS*, que realizam operações em ambientes de trabalho desejados como específicos e perfeitamente estruturados, a *IMC* concentra-se na criação de uma família dinâmica de ferramentas robóticas inteligentes e auto-configuráveis, passíveis de realizarem operações em ambientes de trabalho imprecisos ou com pouca especificidade, com materiais macios, variáveis e flexíveis, típicos do fabrico de calçado [Spencer Jr, 1996].

Uma das abordagens da IMC foi desenvolver um sistema flexível, com equipamentos de sensorização e visão sofisticados, associados a complexos algoritmos, que fazem com que

cada operação realizada em cada sapato seja única. Ao contrário de outras soluções disponíveis no mercado, que apenas asseguram o correcto posicionamento da forma do sapato na célula robotizada, a IMC criou um sistema que contempla a localização do sapato e da forma, uma vez que pode haver variações da posição relativa entre estes (Figura 2.8). Assim, as trajectórias dos robôs são adaptadas a cada caso e uma operação nunca é igual à outra, mesmo para sapatos supostamente iguais.



Figura 2.8 – Stand demonstrativo da *IMC*[11]

2.3 Sistemas robotizados não comerciais

Esta pesquisa bibliográfica incidiu também na pesquisa de sistemas robotizados não comerciais. Estes sistemas são normalmente desenvolvidos por universidades, institutos e laboratórios de pesquisa e desenvolvimento, de fabricantes de soluções robotizadas comerciais e de empresas de automação associadas ao ramo do calçado. As soluções que constroem, integram muitas das vezes, equipamentos comercias ou pretendem replicá-los e optimizá-los, adaptando-os tanto a nível de *software* como de *hardware*, a fim de procurar fazer cumprir o propósito que se predispuseram alcançar.

Seguidamente são apresentados alguns projectos criados por duas universidades e dois institutos, destinados à aplicação na indústria do calçado.

[11] Imagem retirada de Spencer Jr, [Spencer Jr,1996].

“Automatic surface roughing with 3D machine vision and cooperative robot control”

Este é o título de um projecto desenvolvido na Escola de Engenharia Mecânica e de Sistemas da Universidade de Newcastle, no Reino Unido, em que foi concebido um sistema automático de cardagem com identificação prévia do perfil do sapato através de um sistema de visão 3D e geração automática da trajectória.

Devido à complexidade dos diferentes perfis de sapatos existentes, os sensores convencionais não são suficientemente sofisticados e apresentam-se incapazes de calcular a trajectória de desbaste. Assim, como se mostra na Figura 2.9, esta solução da Universidade de Newcastle utiliza um sistema de visão artificial 3D que recorre a uma câmara analógica e a um laser.

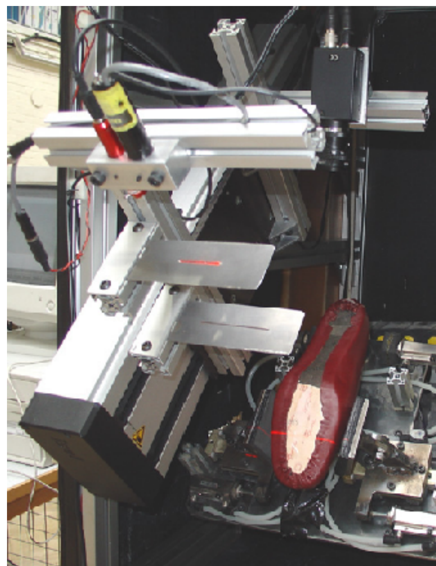


Figura 2.9 – Sistema de visão 3D

Estes são movimentados por acção de um motor passo-a-passo ao longo de uma trajectória linear, permitindo fazer a identificação do perfil da parte superior do sapato, que se encontra preso com uma garra pneumática a uma mesa giratória.

Os resultados obtidos apresentam-se como pontos discretos, com coordenadas definidas em relação a um referencial localizado na estrutura de suporte do sapato.

Esses pontos são utilizados para criar uma curva através de uma interpolação utilizando um método denominado *NURBS - Non-Uniform Rational B-Splines* (Figura 2.10).

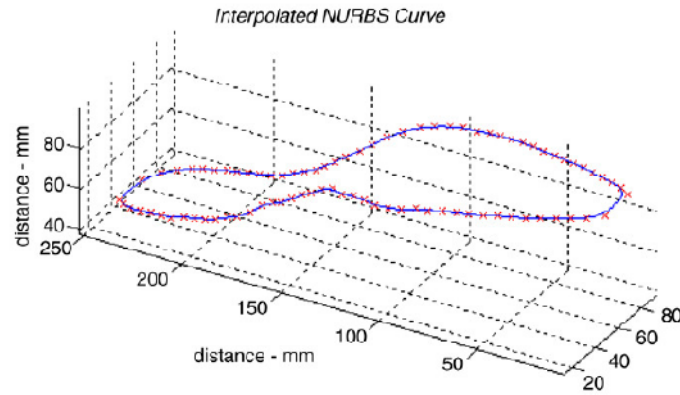


Figura 2.10 – Geração da trajectória pelo método *NURBS*

Após a geração da trajectória, a operação é executada por um robô de seis eixos, recorrendo a um sistema de controlo conjugado que emprega o grau de liberdade da mesa giratória como um grau de liberdade adicional do sistema, e o utiliza com o intuito de minimizar os movimentos do robô.

Como é visível na Figura 2.11, o método de desbaste é realizado por decapagem, com projecção de material granular, num volume fechado.

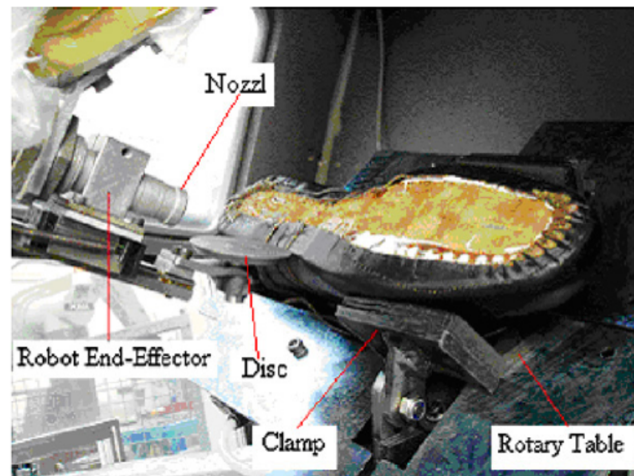


Figura 2.11 – Sistema de desbaste

Este sistema processa esta operação com um erro inferior a 1mm, e demora cerca de 10 segundos para fazer a identificação do perfil através do sistema de visão e 15 segundos para a operação de desbaste com uma velocidade linear de 100mm/s.

Dadas as suas características este sistema apresenta uma elevada eficiência e robustez, permitindo comportar potenciais perturbações.

Célula Automatizada para Operações de Acabamento

Este é o nome de um projecto concebido no Instituto Jožef Stefan na Eslovénia, no qual foi desenvolvida uma célula automatizada para operações de acabamento realizadas em sapatos. Estas operações envolvem o polimento, com aplicação de cera e sprays, e a escovagem.

Após a análise dos processos manuais, os investigadores desta instituição criaram uma célula automatizada com um robô comercial com 6 graus de liberdade da *ABB*, e máquinas dedicadas para o polimento e aplicação de ceras e sprays (Figura 2.12).



Figura 2.12 – Célula automatizada para operações de acabamento

A produção personalizada em massa requer uma preparação prévia de todas as fases produtivas durante a etapa de design de cada sapato, a fim de evitar tempos mortos e quebras na produtividade. Assim utilizaram e desenvolveram ferramentas de CAD para estas operações. Ferramentas essas que possibilitam a geração automática das trajectórias e optimização das mesmas (Figura 2.13), com a previsão de eventuais colisões entre o robô e o ambiente circundante.

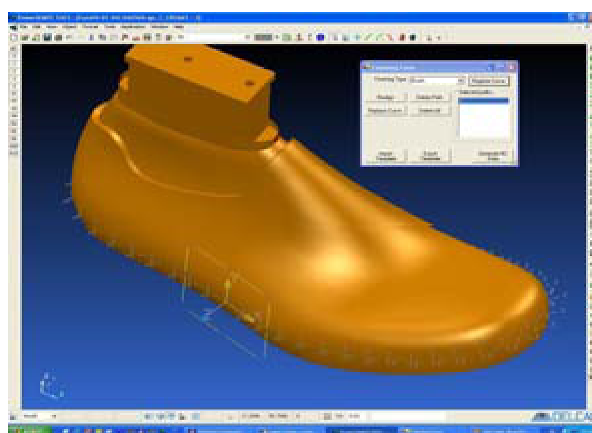


Figura 2.13 – Geração da trajectória a partir de um modelo CAD

Uma vez que a programação e simulação *off-line* do robô pode ser um processo difícil e moroso, a fim de evitar os obstáculos que se impõe no volume de trabalho do robô, a optimização das trajectórias foi baseada na utilização da redundância cinemática do manipulador, fazendo uso da multiplicidade de configurações que este pode adoptar para a realização de uma determinada operação. Note-se que o grau de redundância depende da tarefa que o robô executa.

Sistema Automatizado para Halogenação de Solas de Sapatos

O Departamento de Electrónica Industrial da Universidade do Minho desenvolveu, em 1997, um sistema automático para halogenação da sola de sapatos. A halogenação é um processo químico de limpeza do contorno da sola, que faz com que a superfície adira convenientemente ao sapato aquando da sua colagem. É um processo que tradicionalmente é realizado manualmente por um operador manipulando materiais tóxicos, pelo que é potencialmente perigoso para o operador e daí a necessidade de automação.

Os intervenientes neste processo utilizaram um sistema de visão artificial que, interagindo com um robô, permite a execução desta operação de maneira automática. Para o sistema de visão utilizaram uma pequena câmara de vigilância CCD, uma carta de aquisição de vídeo e um computador. A câmara orientada na perpendicular à sola regista a imagem, que depois é tratada pela carta de aquisição de vídeo e que a converte numa imagem a preto e branco. A imagem é processada e são retirados os pontos dominantes do contorno da sola. Esses pontos são enviado para o controlador do robô através da porta paralela, e o robô, de cinco eixos, move-se depois ao longo desses pontos executando a operação pretendida com um dispositivo construído para o efeito.

O dispositivo de visão e de halogenação, que está fixo ao manipulador, foi construído com recurso a componentes reciclados, como uma bomba de combustível de um automóvel, uma agulha de enchimento de cartuchos de impressão entre outros. O dispositivo encontra-se montado ao lado de uma garra que possibilita também a manipulação da sola (Figura 2.14).

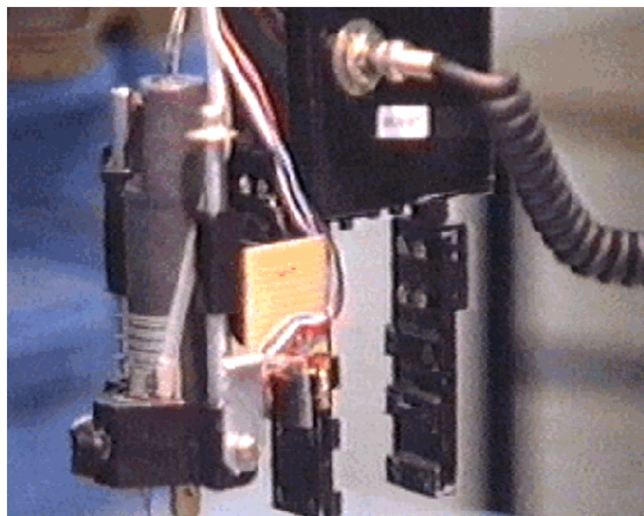


Figura 2.14 – Dispositivo de visão, manipulação e halogenação

Este sistema consegue reduzir em cerca de 50% o tempo necessário para a operação, com melhoria evidente das condições de trabalho para os operadores, que assim não estão sujeitos a atmosferas tóxicas.

***ITIA-CNR* - Célula Robotizada de Cardagem e Colagem para Manufactura de Sapatos Personalizados**

ITIA-CNR é a sigla que representa o *Instituto di Tecnologie Industriali e Automazione – Consiglio Nazionale delle Ricerche*. O objectivo desta organização Italiana é a promoção da inovação industrial, e centra a sua actividade na investigação científica e desenvolvimento tecnológico para a competitividade e sustentabilidade das indústrias de manufactura Italianas e Europeias.

Um dos seus projectos foi realizado no âmbito de um programa europeu de pesquisa na área do calçado, intitulado *EUROSHOE*, tendo desenvolvido um protótipo de uma mini fábrica semi-automatizada para a produção em massa de sapatos personalizados à medida de cada cliente. O processo produtivo nesta fábrica inicia-se com a aquisição de uma imagem tridimensional do pé do cliente, a partir do qual será construído e adaptado o modelo de sapato escolhido. Uma vez obtido o design do produto, este é encaminhado para o departamento de produção que lhe dá a forma desejada. Apesar da especificidade de todo este processo, etapas como a cardagem, a colagem e o polimento são automatizadas.

Para a produção personalizada de calçado este Instituto desenvolveu uma célula automatizada de cardagem dos sapatos e colagem das solas.

Distanciando-se das práticas correntes de geração de trajectórias, baseadas em processos morosos de ensinamento de pontos por parte de operadores especializados, este sistema gera automaticamente as trajectórias directamente a partir de um modelo CAD/CAM, eliminando assim a necessidade de procedimentos prévios de digitalização.

Contudo, a peça real pode divergir do modelo de CAD em vários aspectos, como a falta de uniformidade do cabedal, a falta de precisão dimensional da forma, o incorrecto posicionamento da forma no local de trabalho, e ainda o incorrecto posicionamento do sapato em relação à forma. Assim a trajectória real terá que ser ajustada considerando estas perturbações, podendo divergir significativamente da trajectória gerada em CAD.

A solução desenvolvida centrou-se no uso de um sistema com uma macro-micro abordagem. Por outras palavras, esta abordagem recorre ao uso de um macro manipulador, tipicamente um robô industrial, controlado em posição, e um micro manipulador usado para o controlo de força (Figura 2.15).



Figura 2.15 – Macro e micro manipuladores

Este método oferece uma vantagem considerável, do ponto de vista em que os controladores do macro e do micro manipulador são independentes.

Assim, um robô industrial com 6 graus de liberdade e um sistema robusto de controlo de posição, pode executar a trajectória nominal gerada, ao mesmo tempo que um dispositivo

com um grau de liberdade e com uma maior largura de banda lida com os desvios em relação à trajectória gerada.

O micro manipulador é montado na extremidade do macro manipulador, e contém as ferramentas necessárias para as operações de cardagem e colagem.

A análise deste trabalho, revelou-se particularmente interessante, pois a abordagem adoptada por este instituto e implementada no sistema desenvolvido pode, de certa forma, assemelhar-se ao trabalho que se pretende realizar neste projecto, do ponto de vista do desenvolvimento de um equipamento auxiliar para interagir com um robô, num sistema automatizado de produção.

2.4 Dispositivos de controlo de força

Uma vez que o dispositivo auxiliar que se pretende construir neste projecto será controlado em força, analisaram-se também dois tipos de soluções que permitem implementar o controlo de força. O primeiro envolve o uso de um sensor de força de 6 eixos concebido com o intuito de ser montado directamente na extremidade do braço robótico, sendo depois necessário que o controlador do robô possua *software* adequado para a implementação do controlo de força. O segundo é um dispositivo autónomo passível de ser montado quer no robô, quer numa ferramenta, ou bancada externa, que permite controlar a força de contacto numa única direcção.

Dispositivo baseado no sensor ATI DELTA IP60

Exemplo de um sensor de força comercial é o dispositivo *ATI DELTA IP60*, existente no laboratório de Robótica do Departamento de Engenharia Mecânica (Figura 2.16). Trata-se de um dispositivo capaz de medir seis componentes, três de força e três de binário, que consiste num transdutor equipado com um sistema de aquisição de dados inteligente.



Figura 2.16 – Sensor de força *ATI* DELTA IP60

Este sensor, juntamente com o *software* de controlo do robô tem um custo de aquisição elevado, podendo ascender a valores na ordem dos 25 000 euros. A par do seu elevado custo, acresce o facto de possivelmente esta solução possuir capacidades que estão subaproveitadas na maior parte dos processos produtivos, tornando-se muitas vezes redundante medir forças em mais do que uma direcção.

PushCorp AFD70 Passive Compliance Force Device

Este dispositivo (Figura 2.17) é um exemplo de um sistema uniaxial de controlo de força passivo disponibilizado pela empresa *PushCorp* (USA). A regulação da força é realizada manualmente ou electricamente fazendo variar a pressão de ar comprimido de um actuador pneumático.

Utiliza um cilindro pneumático de baixo atrito com o corpo em vidro e o êmbolo em grafite associado a um sistema de guiamento linear. Dispõe ainda de um sensor de posição interno constituído com um potenciómetro linear, para monitorizar a posição da carruagem. Consegue aplicar uma força até 222N, com uma resolução de +/- 0.5N.



Figura 2.17 – *PushCorp* AFD70

PushCorp AFD1000 Active Compliance Force Device

Este equipamento (Figura 2.18) é idêntico ao anterior contudo tem a capacidade de exercer um controlo de força activo. O seu funcionamento requer um controlador apropriado, também fornecido por esta empresa, e tem a capacidade de detectar automaticamente a carga, e actuar de maneira a compensar as acelerações. Consegue manter uma força constante independentemente da orientação, podendo ser montado na horizontal, na vertical, e invertido na horizontal.

Consegue controlar forças até 222N com uma resolução de +/- 0,4N, segundo um curso linear de trabalho de 20mm.



Figura 2.18 - *PushCorp* AFD1000

Pode ser aplicável a uma ampla gama de processos industriais de maquinagem, polimento, entre outros, que envolvam o controlo de força.

2.5 Conclusão

Após esta pesquisa sobre os sistemas robotizados na indústria do calçado, concluiu-se que várias empresas de automação fornecem já uma gama mais ou menos alargada de soluções genéricas para determinadas etapas produtivas de calçado, bem como projectos de linhas de produção com um elevado nível de automatização, concebidas à medida dos seus clientes.

Os sistemas desenvolvidos por universidades e institutos de pesquisa associados a este ramo, centram-se numa abordagem mais tecnológica, particularizando cada sapato como

sendo um caso. Para isso recorrem normalmente a sistemas dispendiosos de sensorização e controlo, integrando muitas das vezes a visão artificial e a geração automática de trajectórias.

Em qualquer dos casos, pode compreender-se que, a informação disponível não sendo escassa, escasseia no entanto em detalhes e especificidade dos sistemas desenvolvidos, estando mais voltada para a área comercial e não tão focalizada na área científica ou tecnológica.

3 . Arquitectura do Sistema Robótico para Operações de Acabamento

3.1 Introdução

Neste capítulo são estudadas soluções para a concepção da célula robotizada e do dispositivo auxiliar de acabamento a lhe ser integrado, que melhor satisfaçam os requisitos impostos e atenda aos recursos disponibilizados.

Pretende-se, assim, criar uma célula robotizada, recorrendo a um robô controlado em posição e velocidade, e a um dispositivo auxiliar de acabamento controlado em força, concebido para o efeito.

3.2 Concepção da célula robotizada

Tipicamente o conceito de célula robotizada subentende a presença de determinados sistemas. Sistemas esses que normalmente compreendem a existência de um robô e do respectivo controlador, uma máquina dedicada ou uma ferramenta transportada pelo robô, um sistema de alimentação/remoção dos produtos, sistemas de posicionamento, mecanismos de segurança, entre outros.

A célula robotizada que se pretende implementar neste projecto, por questões de logística e tempo, compreende apenas os sistemas básicos, como o robô (e o respectivo controlador) e o dispositivo de acabamento. Os restantes sistemas são objecto de um estudo posterior, não sendo abordados em profundidade, pois não comprometem a problemática essencial deste projecto.

Configuração da Célula Robotizada

Tipicamente, uma célula robótica apresenta uma das seguintes configurações:

- **Robô centrado na célula**

Nesta configuração o robô encontra-se fixo ao solo, ou suspenso num pórtico, e dispõe-se no centro da célula robotizada, com os restantes equipamentos auxiliares de produção e transporte dispostos em seu redor.

- **Robô em linha com o sistema de movimentação**

Nesta configuração, o robô encontra-se fixo ao solo ou suspenso, e em linha com o sistema de transporte da célula. Normalmente esta configuração incorpora a utilização de vários robôs ao longo da mesma linha.

- **Robô móvel na célula**

Nesta disposição o robô pode movimentar-se ao longo da célula para executar operações em diferentes máquinas. Torna-se necessário no entanto fazer um estudo do número de máquinas que o robô pretende servir, bem como a sua organização dentro da célula, a fim de optimizar ritmos de produção e evitar tempos mortos na sua deslocação.

Sistema de Transporte

Quanto à sua configuração física, o sistema de transporte das peças/produtos, pode apresentar-se como sendo em linha ou fila, em carrossel, em mesa giratória ou de indexagem, e ainda em paralelo. A configuração adoptada deve ser alvo de um estudo prévio, e estar adaptada às etapas produtivas de cada peça, maximizando as cadências de produção, utilizando o mínimo de recursos possível.

Quando à temporalidade do sistema de transporte, este pode ainda apresentar-se com sendo [Abreu, 2001]:

- **Contínuo -** Em que os produtos são movidos a uma velocidade constante, e as operações são realizadas com os produtos em movimento. Assim, só algumas operações simples são possíveis de implementar;
- **Síncrono ou intermitente -** Em que os subprodutos ou produtos são transportados em simultâneo, intermitentemente, entre as várias estações de montagem;
- **Assíncrono -** Em que cada subproduto ou produto é transportado independentemente, logo que acabe de ser realizada a operação prevista na estação de trabalho onde se

encontrava. Estes sistemas oferecem, assim, uma maior flexibilidade que os outros dois sistemas anteriores.

Robô

O robô utilizado neste projecto é o *ABB IRB2400*, disponível no laboratório de Robótica do Departamento de Engenharia Mecânica da Faculdade de Engenharia da Universidade do Porto, com o respectivo controlador *IRC5* (Figura 3.1). Este manipulador de seis eixos, tem um alcance máximo de 1.5m, uma capacidade de carga máxima de 20kg e uma repetibilidade de 0.06mm.

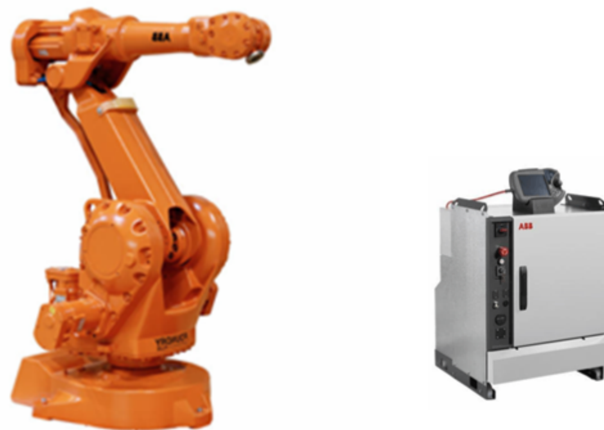


Figura 3.1 – Robô *ABB IRB2400* e controlador *IRC5*

Suporte do Sapato

Apesar de ter um papel bastante relevante, a fixação das formas que contém os sapatos, não será objecto de uma análise neste trabalho, podendo ser elaborado um estudo posterior, para a optimização deste parâmetro.

Dispositivo de Controlo da Força de Contacto

O que se pretende do dispositivo de acabamento, para além deste poder suportar o sapato, ou uma ferramenta de corte/desbaste que permita a realização da operação de acabamento, removendo o material em excesso da sola, é que possua também um sistema de controlo de força, que permita a regulação da força de contacto entre o sapato e a ferramenta.

Atendendo ao equipamento disponível no laboratório de robótica, da Faculdade de Engenharia da Universidade do Porto, adoptou-se a configuração de robô centrado na célula (Figura 3.2), por ser a configuração mais adequada às condições existentes.

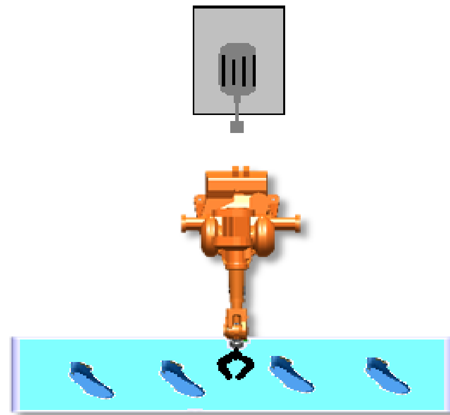


Figura 3.2 – Robô centrado na célula robotizada

3.3 Estudo de soluções para o dispositivo de acabamento

O dispositivo auxiliar de acabamento que se procura construir, pretende-se que seja capaz de controlar a força de contacto entre a peça e a ferramenta. Dependendo da sua natureza construtiva, pode servir de suporte à ferramenta, ou à peça, e deve ainda ter a capacidade de comunicar com o controlador do robô, estando a informação disponível para ser utilizada no controlo dos movimentos do robô (quer em posição, quer em velocidade da trajectória), ou para controlar a ferramenta.

Localização do Dispositivo de Acabamento

As operações de maquinagem leve com robôs envolvem, tipicamente, a utilização do robô que transporta uma ferramenta a fim de executar o trabalho sobre o produto, ou nalguns casos, o robô limita-se a posicionar o produto em relação à ferramenta que se encontra fixa.

Assim, existem duas configurações possíveis para a localização do dispositivo de acabamento:

- **Dispositivo transportado pelo robô e sapato fixo na linha de transporte**

Nesta configuração, mostrada na Figura 3.3, o dispositivo de acabamento, com a respectiva ferramenta, é transportado pelo braço do robô e descreve a trajectória adequada a fazer o acabamento do sapato que se encontra fixo na linha de transporte da célula.

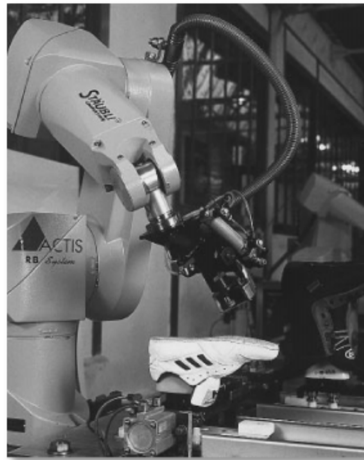


Figura 3.3 – Dispositivo fixo ao robô

- **Dispositivo fixo em bancada e sapato transportado pelo robô**

Nesta configuração (Figura 3.4), o dispositivo de acabamento, com a respectiva ferramenta, está fixo numa bancada, e é o sapato quem se desloca, movimentado pelo robô.



Figura 3.4 – Dispositivo fixo em bancada

Destas duas possíveis configurações, a primeira apresenta uma clara vantagem em termos do tempo necessário para executar a operação pretendida, pois ao transportar a ferramenta, o robô não necessita de retirar o sapato da linha de alimentação/remoção da célula, o que se traduz em importantes ganhos temporais no processo. É necessário no entanto que o sistema de transporte possua um sistema de posicionamento preciso do sapato em relação ao robô, e com boa repetibilidade.

A segunda configuração apresenta a vantagem de exigir menos capacidade de carga ao robô e, possivelmente, um sistema de acabamento mais simples. No entanto implica a

existência de um sistema de fixação que garanta uma correcta orientação da forma em relação ao elemento terminal do robô.

Ferramenta de Corte

Actualmente não existe muita informação sobre a maquinagem de elastómeros. Tradicionalmente e para este caso em particular, o acabamento da sola faz-se com uma máquina dedicada com um sistema de lixa de fita.

Apesar da informação disponível ser muito escassa, a maquinagem por fresa cilíndrica apresenta-se como uma opção bastante viável, com algumas vantagens significativas de aplicabilidade neste projecto, em relação às outras opções consideradas, e apresentadas na Tabela 3.1.

Ferramenta	**Vantagens**	**Desvantagens**
Lixa de fita	-Bom acabamento	-Atravancamento
Mó abrasiva	-Bom acabamento	-Acumulação de impurezas na mó
Fresa cilíndrica	-Reduzido atravancamento -Facilidade de troca de ferramenta	-Aquecimento da ferramenta

Tabela 3-1 – Ferramentas de corte/desbaste

Assim, foi definido que a ferramenta de corte utilizada seria uma fresa cilíndrica, dadas as suas vantagens, anteriormente mencionadas.

Parâmetros de Corte

A correcta definição dos parâmetros de corte é de importância fulcral na qualidade final apresentada pelo sapato acabado. Contudo a operação de acabamento deste tipo de sapatos consiste na remoção de material por maquinagem da sola, que neste caso é de borracha.

Na sua maioria as peças em elastómeros como a borracha, são obtidas por moldação e não por maquinagem. Por este motivo a literatura disponível sobre este assunto é muito

escassa, concentrando-se na divulgação de resultados de trabalhos experimentais muito específicos [Santos, 2008].

A maquinagem de elastómeros com boa precisão dimensional pode tornar-se um processo difícil, dadas as características deste material. Alguns autores de trabalhos de pesquisa sugerem o aquecimento da ferramenta, enquanto outros sugerem o arrefecimento criogénico do material [Shih, 2004]. Dada a aparente inexistência de ferramentas apropriadas para estes processos, normalmente as ferramentas de corte usadas são fresas para madeira.

- **Diâmetro e número de dentes**

O diâmetro e o número de dentes da ferramenta, são dois factores a ter em atenção na escolha da ferramenta. Ferramentas com diâmetro reduzido, têm uma velocidade periférica menor o que implica uma velocidade de rotação maior da ferramenta, ao passo que o aumento excessivo do seu diâmetro pode não assegurar um acabamento desejado em peças com contornos apertados. O aumento do número de dentes possibilita um melhor acabamento, mas permite também a acumulação de material entre os gumes cortantes, resultando num aquecimento indesejado da ferramenta, e um corte defeituoso.

Assim, escolheu-se genericamente uma fresa com 20mm de diâmetro com três dentes rectos.

- **Sentido de corte**

A definição do sentido de corte depende do material a maquinar, e tem influência directa no acabamento final apresentado. Dadas as características elásticas do material a maquinar, optou-se por uma fresagem discordante em detrimento da fresagem concordante (Figura 3.5).

Neste tipo de fresagem, o ataque da fresa à peça inicia-se tangencialmente. O arranque da apara e a evolução das forças de corte ocorrem de uma maneira progressiva ao contrário da fresagem concordante.

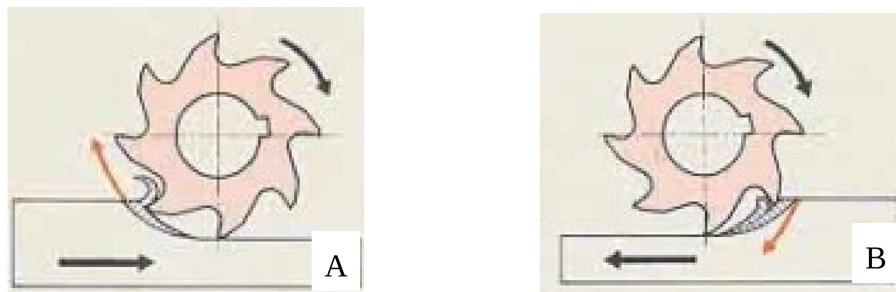


Figura 3.5 – Fresagem cilíndrica discordante (A) e concordante (B) [12]

- **Velocidade de corte e velocidade de avanço**

Apesar da sua importância extrema para o cálculo da potência de corte, dada a inexistência de valores particulares para este processo em questão, utilizaram-se valores indicativos de velocidade de corte e avanço utilizadas em processos de fresagem de nylon. Esses valores são 1000 m/min (16,67m/s) para a velocidade de corte, e 0,2mm/rotação para o avanço[13].

- **Potência de corte**

Mais uma vez, dada a ausência de informação específica, a potência de corte foi calculada com recurso ao mesmo método que é utilizado na fresagem dos metais, partindo da equação seguinte.

$$P_c = F_c * V_c \quad (1)$$

P_c – potência de corte	[W]
V_c – velocidade de corte (16,67m/s)	[m/s]
F_c – força de corte	[N]

Para o cálculo da força de corte é necessário conhecer o valor da pressão específica de corte. A pressão específica de corte não é mais do que a força de corte por unidade de área da secção a maquinar, e depende das características do material e da velocidade de corte, entre outros factores. À falta de valores específicos deste parâmetro para elastómeros, foi utilizado o valor correspondente para a ebonite. No entanto a ebonite é um tipo de borracha com

[12] Imagem retirada de www.engprod.ufjf.br/processos/USINAGEM/FRESAMENTO_arquivos/image007.jpg em 8/1/2010

[13] Dados retirados do site da empresa *Arvy Plast* referentes à fresagem do nylon (www.arvyplast.com.br/novo/produtos3.php?id=74&rqt=Plasticos_Industriais) em 10/01/2010

elevada rigidez e sem elasticidade, pelo que se utilizou um valor de pressão específica corte deste material para velocidades de avanço por rotação superior à definida. Esse valor é de 250N/mm^2 para um avanço de 0,4mm por rotação[14].

$$F_c = K_c * s \tag{2}$$

K_c – pressão específica de corte [N/mm^2]

s – secção de corte [mm^2]

$$s = h * a_c \tag{3}$$

h – largura de corte (altura da sola = 15 mm) [mm]

a_c – avanço de corte [mm]

$$a_c = a_d * \sin \mathbf{f} \tag{4}$$

a_d – avanço por dente [mm]

$\mathbf{f}$ – ângulo entre a direcção do avanço e a direcção de corte [graus]

$$a_d = a_v / z \tag{5}$$

a_v – avanço por volta [mm]

z – número de dentes (3)

$$\mathbf{f} = \arccos (1 - 2 * a_v/D) \tag{6}$$

a_v – espessura do material a remover (1mm) [mm]

D – diâmetro da ferramenta [mm]

Resolvidas as equações, (1 a 6), chegou-se a um valor de cerca de 109 N para a força de corte, e 1,9 kW para a potência de accionamento da ferramenta à velocidade de 16000 rpm.

---

[14] Valor consultados das tabelas de normas alemãs AWF – Ausschuss für Wirtschaftliche Fertigung

Accionamento da Ferramenta

Os meios potencialmente mais adequados para a transmissão de movimento à ferramenta, são o accionamento pneumático e o accionamento eléctrico.

Por razões de ordem energética, económica, e dada a ampla gama de oferta de motores eléctricos para accionamentos industriais, o accionamento eléctrico destaca-se em relação ao accionamento pneumático, pelo que possivelmente este será o tipo de accionamento mais indicado. Contudo, por razões de custo, acabou por se optar por um accionamento pneumático, dada a disponibilidade laboratorial do equipamento existente.

É de realçar o facto de a ferramenta pneumática utilizada ter uma potência de accionamento significativamente mais reduzida do que a potência calculada na definição dos parâmetros de corte. Chama-se a atenção ainda para o facto de esta opção ter apenas influência na qualidade do acabamento final apresentando, não interferindo ou invalidando os pressupostos assumidos na arquitectura global do sistema robotizado para as operações de acabamento. Assim, esta limitação será considerada aquando da realização dos ensaios experimentais de validação do sistema de acabamento.

Controlo de Força

A capacidade do dispositivo regular a força de contacto entre o sapato e a ferramenta, é um dos principais requisitos do sistema a implementar. Pretende-se que a sua construção seja simples, económica e adequada à utilização de um robô controlado em posição e velocidade.

Estudadas as configurações genéricas possíveis para o dispositivo de acabamento, e uma vez que na realização das operações de acabamento das solas dos sapatos pretende utilizar-se um robô com 6 graus de liberdade, foi idealizado um dispositivo de acabamento auxiliar fixo, exterior ao robô, que serve de suporte à ferramenta. O robô fica responsável pela manipulação do sapato.

A força de contacto que se pretende controlar é exercida na direcção perpendicular à superfície a maquinar. Assim, tirando partido da elevada capacidade de posicionamento/orientação e a multiplicidade de configurações disponíveis que um robô deste tipo possui, pode-se projectar um sistema de controlo de força de reduzida complexidade.

Desprezando-se as forças geradas pelo processo de corte, uma direcção de controlo de força é suficiente para este sistema, desde que o robô assegure a perpendicularidade da sola do sapato em relação à direcção segundo a qual a força é controlada.

O dispositivo possuirá então um grau de liberdade linear segundo o qual é exercido o controlo da força.

Como se pode ver na Figura 3.6, a força de contacto deverá ser controlada na direcção perpendicular à superfície a maquinar. O robô será responsável por assegurar continuamente o posicionamento do sapato, orientando-o correctamente em relação à ferramenta, justificando-se assim a necessidade de o dispositivo possuir apenas uma direcção de controlo de força.

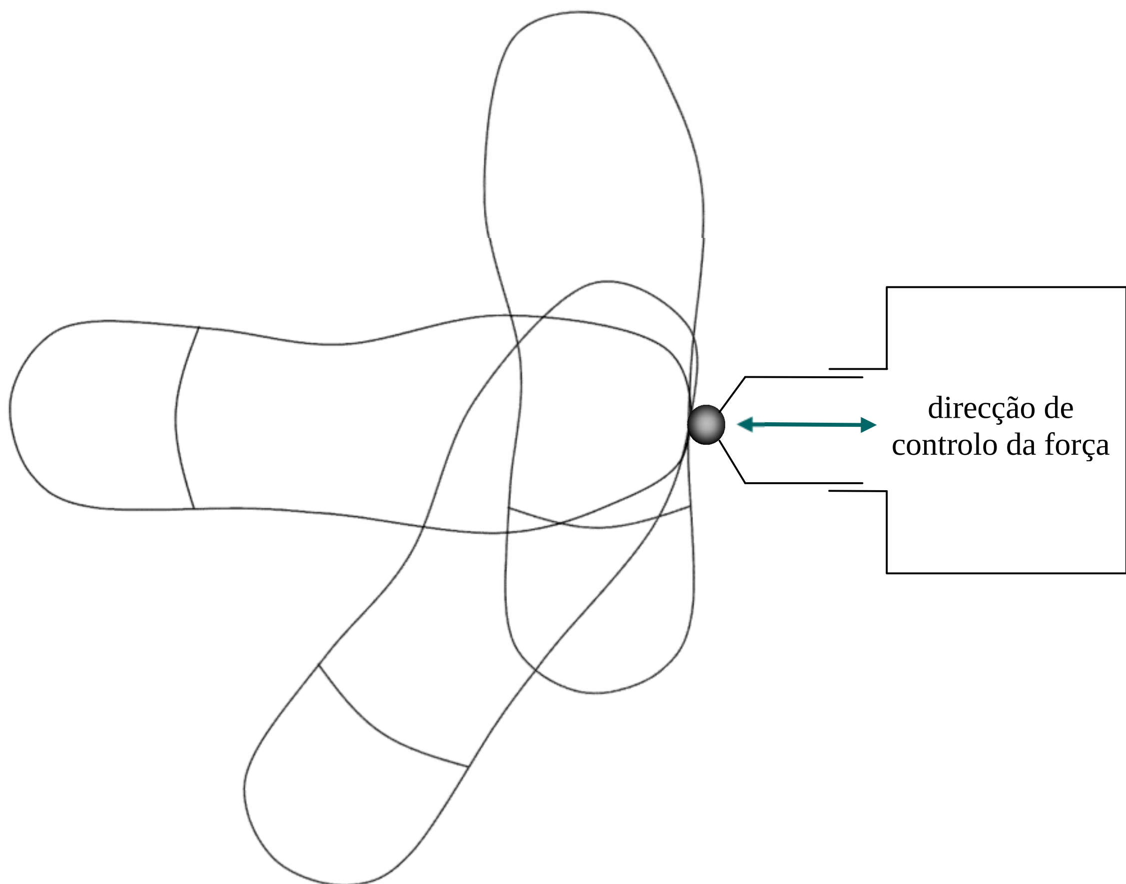


Figura 3.6 – Direcção de controlo da força

Com isto, pretende-se eliminar a necessidade da existência de um sistema de controlo de força no braço robótico, consideravelmente dispendioso, com potencialidades possivelmente subaproveitadas para o processo em causa.

Medição de Deslocamento e Comunicação

Dado que o sistema de controlo de força que se pretende instalar no dispositivo implicará a ocorrência de um deslocamento da ferramenta em relação à sua posição inicial de referência, torna-se necessário medir este deslocamento. Assim será possível processar os dados da posição, e ajustar o controlo do movimento do robô ou da ferramenta.

Ao contrário da maior parte das soluções comerciais, pretende-se então construir um sistema de controlo em malha fechada, em que o dispositivo de acabamento comunique com o controlador do robô e este possa modificar os seus movimentos em função do sinal que recebe.

3.4 Princípio de funcionamento do sistema de acabamento

O princípio de funcionamento do sistema que se pretende implementar, está dependente das propriedades do dispositivo auxiliar de acabamento. Assim, torna-se necessário definir primeiramente as suas características funcionais.

Caracterização do dispositivo

Na configuração idealizada, o dispositivo auxiliar encontra-se fixo, no espaço de trabalho do robô, e possui duas características fundamentais:

- Permite a regulação da força de contacto entre a ferramenta que transporta, ao longo de um grau de liberdade linear, e o sapato que é manipulado pelo robô;
- Permite a medição do deslocamento linear correspondente à posição da ferramenta.

A regulação da força é realizada de uma maneira passiva, por meio de um sistema pneumático. O deslocamento é medido com recurso a um codificador incremental linear (*encoder*), e depois de processado, o sinal é enviado para o controlador do robô através da porta série, utilizando os princípios de comunicação via RS-232.

O dispositivo possui ainda esbarros mecânicos ajustáveis, para permitir a limitação do curso do eixo de posicionamento da ferramenta.

Princípio de funcionamento do sistema

Na concepção planeada, o robô controlado em posição e velocidade da trajectória, transporta o sapato que será maquinado no dispositivo auxiliar fixo, exterior ao robô. O robô manipula o sapato de modo a garantir a perpendicularidade da sola com a ferramenta de corte.

Pretende-se que o robô descreva uma trajectória conhecida, com uma velocidade predefinida, percorrendo um conjunto de pontos, correspondentes à forma do contorno final desejado, da sola do sapato.

Durante a execução do movimento de corte, a descrição da trajectória é tangencial à ferramenta de corte. Esta mantém-se na sua posição de referência (definida pelo ajuste de um batente mecânico), sempre que o contorno real da sola do sapato se assemelhe geometricamente à sua forma desejada, imposta pela trajectória (Figura 3.7).

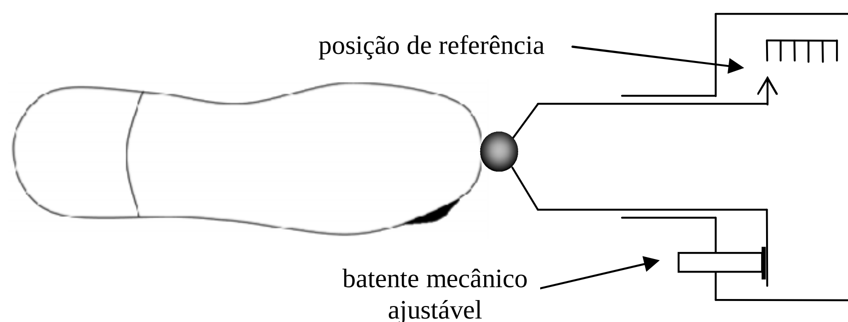


Figura 3.7 – Posição de referência da ferramenta

Nestas situações, a velocidade de execução da trajectória será mantida constante, uma vez que não ocorre o deslocamento da ferramenta.

Quando existir uma diferença entre a cota real do contorno da sola do sapato e a cota desejada, descrita pela trajectória do robô, a ferramenta suportada pelo dispositivo pode deslocar-se da sua posição de referência (Figura 3.8). A diferença entre cotas traduz-se numa sobreespessura de maquinagem, correspondente ao material que é necessário remover.

Durante o deslocamento da ferramenta, a força de contacto é mantida constante, através da utilização de um actuador pneumático alimentado a uma pressão constante.

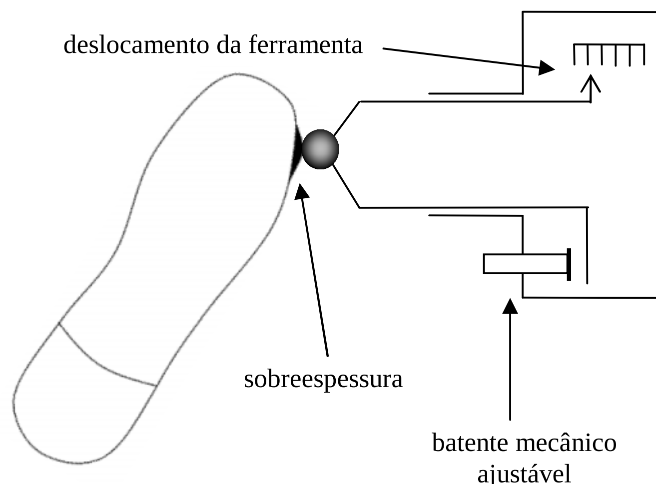


Figura 3.8 – Deslocamento da ferramenta

Nestas situações, a velocidade predefinida da trajectória, é actualizada (reduzida) pelo controlador do robô, em função do sinal recebido da posição da ferramenta.

Assim, a velocidade predefinida da execução da trajectória, será continuamente actualizada em função da maior ou menor quantidade de material a remover.

O esquema da Figura 3.9 mostra os componentes, e o princípio de funcionamento do sistema que se pretende implementar e validar.

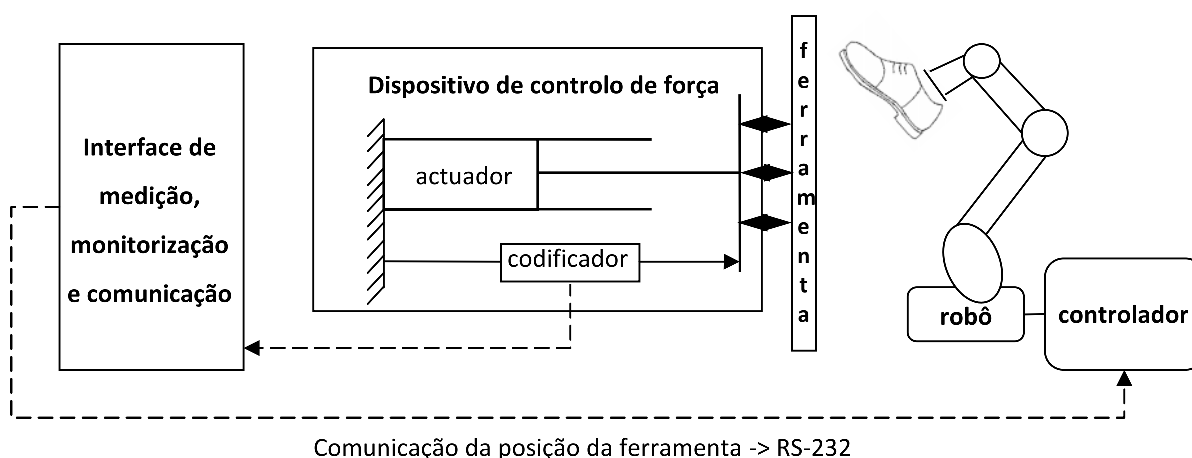


Figura 3.9 - – Princípio de funcionamento do sistema

Comparação do sistema com soluções comerciais existentes

Feita a caracterização do princípio de funcionamento do sistema desenvolvido, tornou-se inevitável a sua comparação com as soluções robotizadas comercias, que integram

dispositivos de controlo de força associados a funcionalidades genéricas do *software* de programação.

Dado que o robô utilizado neste projecto é um *ABB*, foram avaliadas e comparadas as soluções comerciais disponibilizadas por este fabricante, para a realização de processos de maquinagem com recurso ao controlo de força.

Explicam-se então sucintamente os dois processos de controlo de força disponíveis no controlador da *ABB.*

A solução de controlo de força que a *ABB* possui para os seus robôs industriais, recorre a um sensor de força de 6 eixos montado no elemento terminal do robô, e permite implementar dois modos distintos de controlo de força: o *FCPressure* e o *FCSpeedchange*.

No primeiro modo, o *FCPressure,* é definida uma trajectória de referência, uma velocidade de execução da trajectória para o robô e uma referência de força (numa direcção). O robô executa a trajectória definida, podendo alterá-la de modo a manter a força constante no valor predefinido (Figura 3.10). Assim, não é garantida a trajectória de referência, mas sim imposta uma força de contacto que é procurada manter constante.

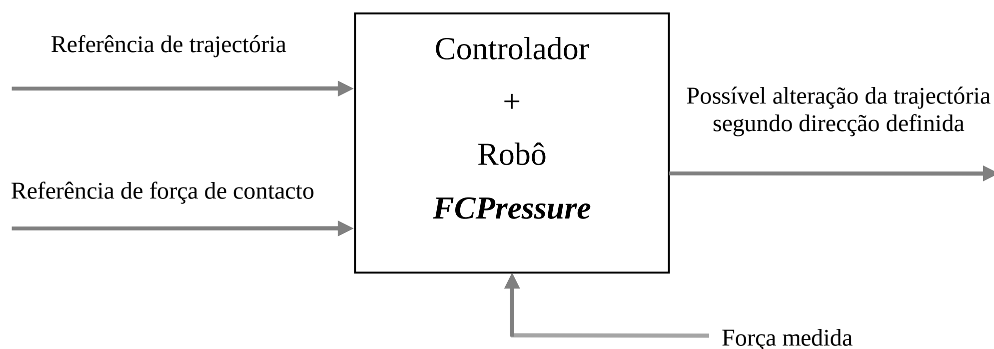


Figura 3.10 – Funcionamento do *FC Pressure*

No segundo modo, o *FCSpeedChange*, é definida uma trajectória de referência, uma velocidade de execução para o robô e definida uma referência de força. O robô executa a trajectória definida, podendo alterar a sua velocidade de execução da trajectória, se a força medida exceder o valor predefinido. Se a força medida for demasiado elevada, é possível programar a modificação da trajectória de referência, ou abortar a operação (Figura 3.11).

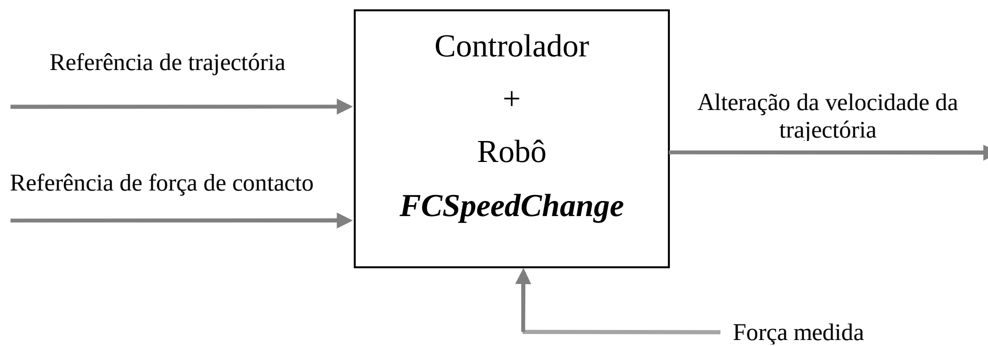


Figura 3.11 – Funcionamento do *FC SpeedChange*

Procedendo a uma esquematização análoga à utilizada para descrever os dois processos anteriores disponíveis no controlador do robô *ABB*, representa-se na Figura 3.12 o funcionamento do sistema desenvolvido.

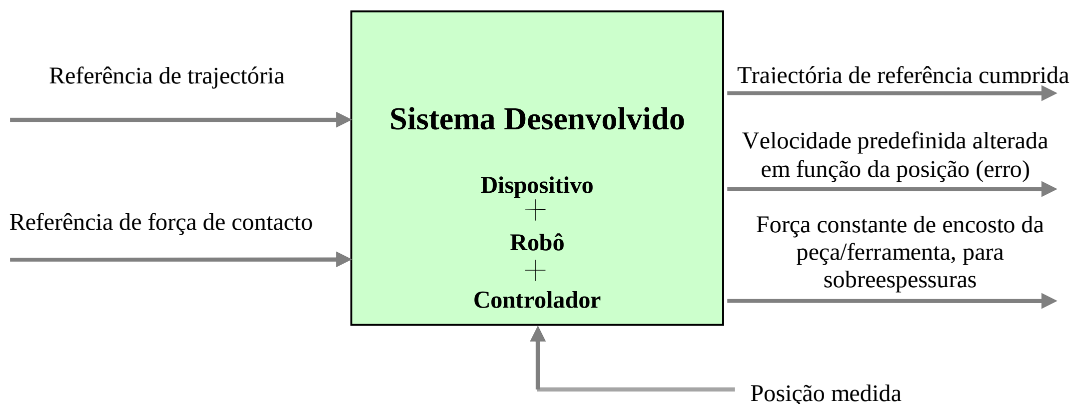


Figura 3.12 – Funcionamento do sistema desenvolvido

Assim, o dispositivo desenvolvido, quando integrado com um robô apenas controlado em posição e velocidade, contempla características semelhantes às existentes no controlador de um robô equipado com controlo de força (com as funções *FCPressure* e *FCspeedChange*), embora limitado a uma só direcção de controlo.

De notar, que apesar de o funcionamento pretendido deste sistema ser o representado na Figura 3.12, o dispositivo pode ainda ser utilizado em dois outros modos distintos, sendo um deles totalmente em controlo de força e outro num controlo cooperativo entre a força e a velocidade da trajectória.

O modo de controlo cooperativo de força e velocidade, é muito semelhante ao modo de funcionamento que se pretende implementar, diferindo apenas na medida em que a força de contacto constante entre a peça e a ferramenta se mantém durante todo o processo de maquinagem, e não apenas quando a ferramenta se acomoda para comportar as sobreespessuras, ou desvios geométricos da sola em relação à sua forma ideal.

No modo de controlo de força pode descartar-se a parte da comunicação e da actualização da velocidade, e fazer a maquinagem de uma peça assegurando uma força de contacto constante entre a peça e a ferramenta, sendo este modo de funcionamento, análogo ao *FC Pressure.*

Em ambos estes modos de funcionamento, para assegurar o contacto permanente entre a sola e a ferramenta, é apenas necessário alterar a posição, real ou virtual, do eixo da ferramenta. Esta alteração pode ser feita a nível mecânico actuando no dispositivo, ou a nível do *software*. Para isso basta actuar sobre o mecanismo fazendo deslocar a ferramenta em relação ao referencial previamente definido, ou definir o referencial virtual, correspondente ao eixo de rotação da ferramenta, com um determinado afastamento em relação ao seu eixo real. Esse afastamento deverá ser definido de maneira a garantir o contacto permanente entre a ferramenta e o sapato.

3.5 **Conclusão**

Neste capítulo foram estruturados os parâmetros gerais do dispositivo de acabamento e esboçado o princípio de funcionamento do sistema de acabamento das solas dos sapatos.

Apesar da sua extrema importância, dada a carência de literatura específica, a definição dos parâmetros de corte foi realizada com base nos procedimentos utilizados na fresagem de metais, em que se fizeram algumas analogias entre o material a maquinar e o nylon e a ebonite, a fim de serem determinadas algumas variáveis.

Foi idealizado um sistema que integra um dispositivo de acabamento exterior ao robô, fixo numa bancada, que transporta a ferramenta de corte, e tem a capacidade de exercer o controlo de força numa direcção, podendo a ferramenta deslocar-se segundo essa direcção.

A posição da ferramenta é medida e comunicada através da porta série para o controlador do robô, sendo a velocidade de execução da trajectória do robô ajustável, em função do valor da posição.

De notar que a escolha do processo de acabamento e da ferramenta de corte terão influência significativa na qualidade do processo a implementar. Contudo, as limitações económicas e os equipamentos existentes ditaram a escolha de um sistema com recurso a uma fresa para madeira e um accionamento pneumático, com características discordantes das consideradas após a determinação dos parâmetros de corte. No entanto, o desenvolvimento de um sistema de controlo de força de contacto é aplicável a uma variedade de processos de acabamento e respectivas ferramentas.

4 . Desenvolvimento de um Dispositivo de Controlo de Força

4.1 Introdução

A estratégia a adoptar na implementação do sistema de acabamento das solas dos sapatos, passa pela construção e integração na célula robotizada, de um dispositivo de controlo de força, idealizado neste projecto atendendo aos requisitos do mesmo e analisados previamente.

Pretende-se assim que o dispositivo de controlo de força possua as seguintes características:

- Suporte para uma ferramenta de acabamento;
- Capacidade da regulação da força de contacto entre a peça e a ferramenta numa única direcção;
- Capacidade de exercer forças de contacto até 50N;
- Curso de trabalho de 50mm;
- Capacidade de regulação da posição de trabalho da ferramenta;
- Medição e visualização da posição de trabalho da ferramenta;
- Capacidade de comunicação de dados da posição através de ligação por porta série.

4.2 Estrutura mecânica

Atendendo à necessidade de o dispositivo de acabamento possuir um grau de liberdade linear, a sua construção desenrola-se fundamentalmente em torno de um elemento principal que é uma mesa linear. Por razões de disponibilidade laboratorial, a mesa utilizada foi o modelo *SGO 12 – 85* da *Rexroth Star* (Figura 4.1). Esta mesa apresenta características compatíveis com a aplicação pretendida, possuindo um baixo atrito, um curso de 165mm e uma capacidade de carga adequada.

Para um valor nominal de deslocamento, esta mesa apresenta uma capacidade de carga máxima de 1500N e um momento máximo de 57Nm segundo a direcção transversal ao movimento e 52Nm segundo a direcção do movimento.

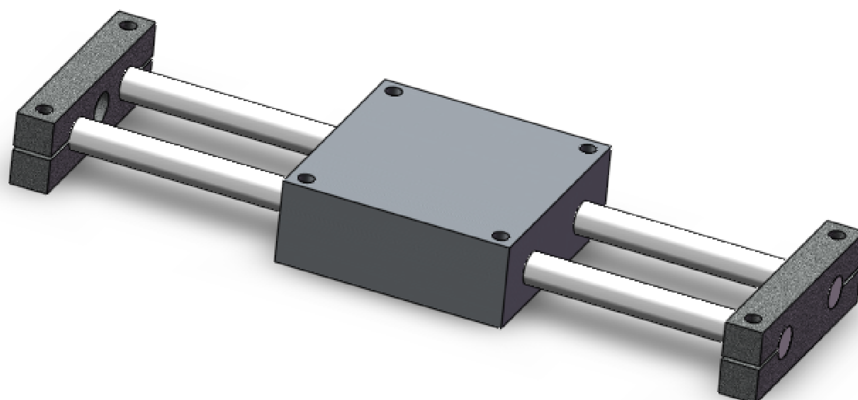


Figura 4.1 – *Rexroth Star SGO 12 - 85*

Foi adicionada uma base de suporte e dois espaçadores aos montantes da mesa. O curso da mesa foi ajustado para o curso de um actuador pneumático, responsável pela regulação da força, e este foi montado num dos montantes desta, num furo roscado M12x1, previamente maquinado.

Os espaçadores permitem uma montagem numa posição mais elevada da mesa e a possibilidade de incorporação de componentes sob esta. O espaçador anterior alberga um parafuso que serve de batente e regula o fim de curso da mesa. Sob a parte móvel da mesa foram montadas estruturas de suporte ao codificador incremental, à respectiva fita e ao batente de ajuste da posição de referência ou de trabalho (Figura 4.2).

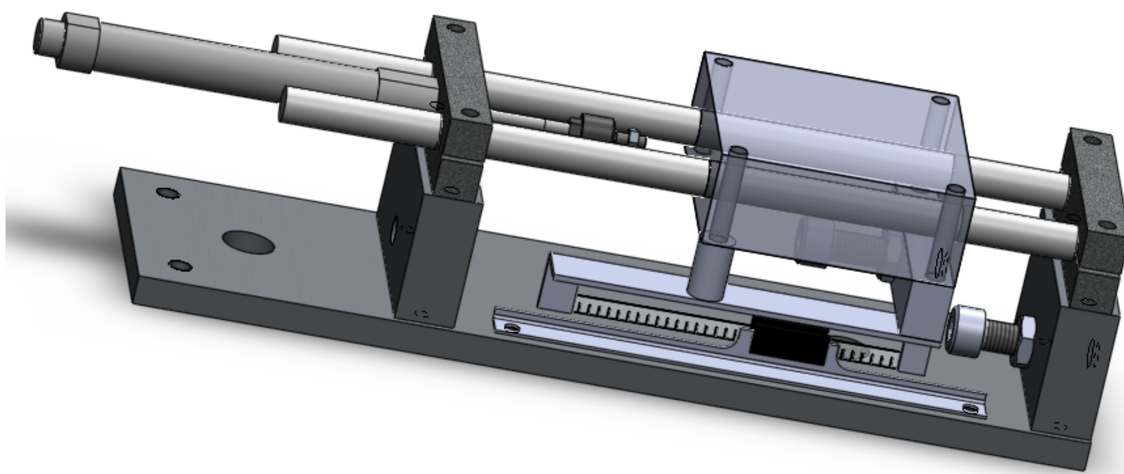


Figura 4.2 – Estrutura do dispositivo de controlo de força

A Figura 4.3 mostra a estrutura porta ferramenta, composta por duas placas unidas com dois esquadros. Os esquadros apresentam dois orifícios circulares com 40mm de diâmetro, que simultaneamente ao aligeirar o peso da peça, possibilitam o controlo visual do ajuste dos batentes de fim de curso. A estrutura porta ferramenta é posteriormente aparafusada à corrediça da mesa, e ligada à haste do cilindro. São adicionadas lateralmente duas placas de acrílico para proteger o *encoder* e a fita, de poeiras e resíduos resultantes do processo de acabamento. A ferramenta é fixa com dois grampos roscados nas extremidades.

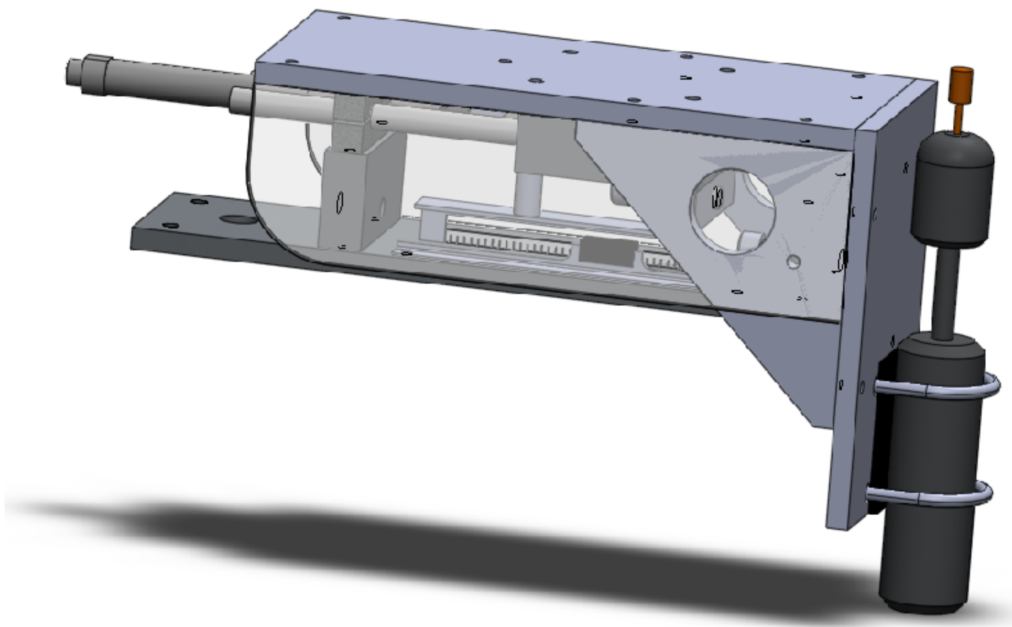


Figura 4.3 – Dispositivo de controlo de força com ferramenta de corte

A modelação da ferramenta efectuada, foi realizada com base num equipamento pneumático existente no laboratório de robótica da Faculdade de Engenharia da Universidade do Porto. As características desta ferramenta não contemplam os parâmetros de corte calculados anteriormente, contudo a estrutura porta ferramenta foi concebida de modo a poder comportar outro tipo de ferramentas, mais adequadas ao processo em questão.

A Figura 4.4 mostra uma vista explodida do dispositivo. A Tabela 4.1 identifica os principais componentes do dispositivo.

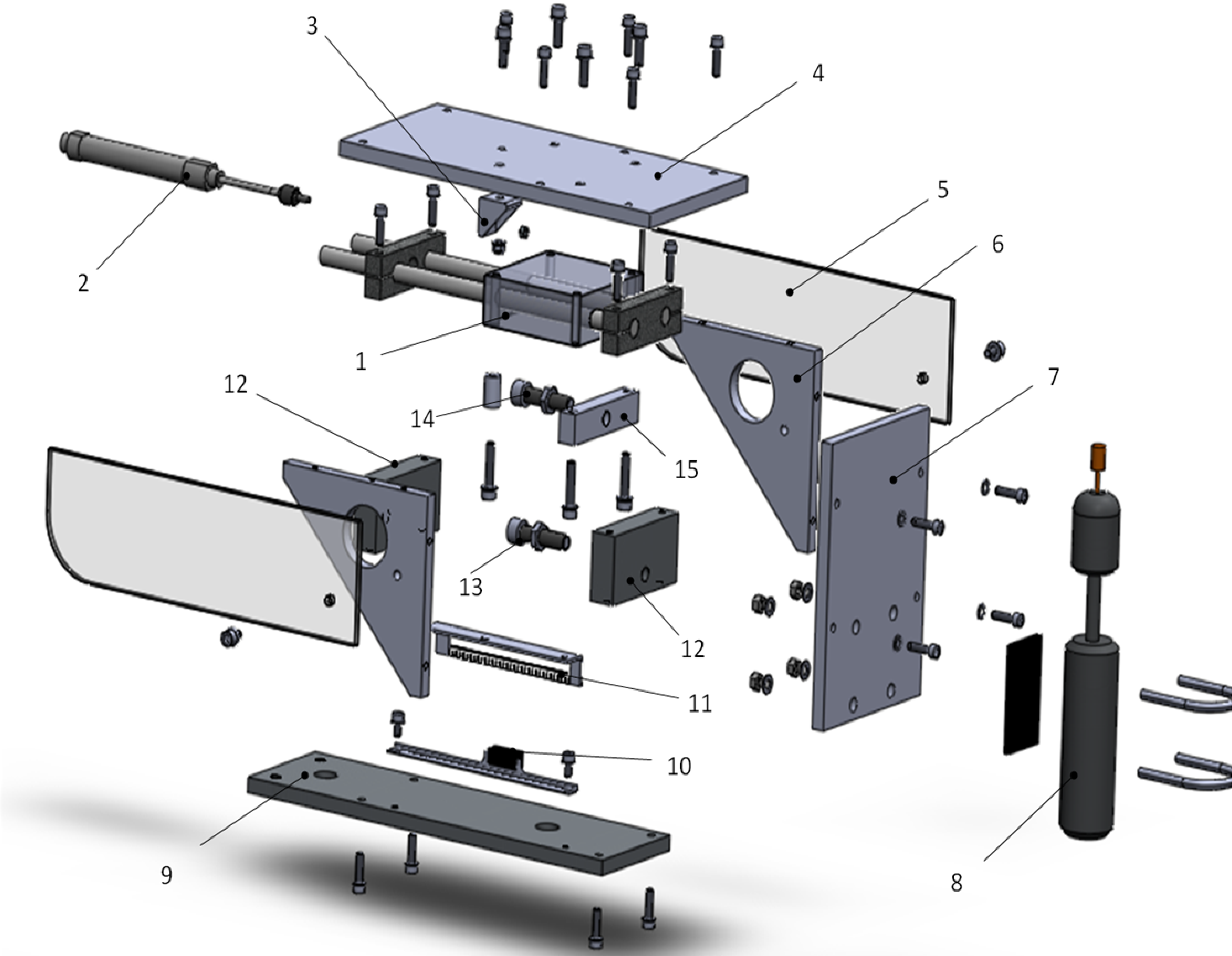


Figura 4.4 – Vista explodida do dispositivo

Nº.	Componente
1.	Mesa linear *Rexroth Star*
2.	Actuador pneumático
3.	Suporte de fixação da haste do actuador
4.	Placa horizontal
5.	Protecção em acrílico
6.	Esquadro
7.	Placa vertical
8.	Ferramenta pneumática
9.	Base do dispositico
10.	*Encoder*
11.	Fita do *encoder*
12.	Espaçador
13.	Batente (ajuste do fim de curso)
14.	Batente (ajuste da posição de referência da ferramenta)
15.	Suporte do batente

Tabela 4-1 – Principais componentes do dispositivo de acabamento

Após a modelação tridimensional do dispositivo e a maquinagem dos componentes necessários, procedeu-se à sua construção. A Figura 4.5 evidencia o dispositivo construído.

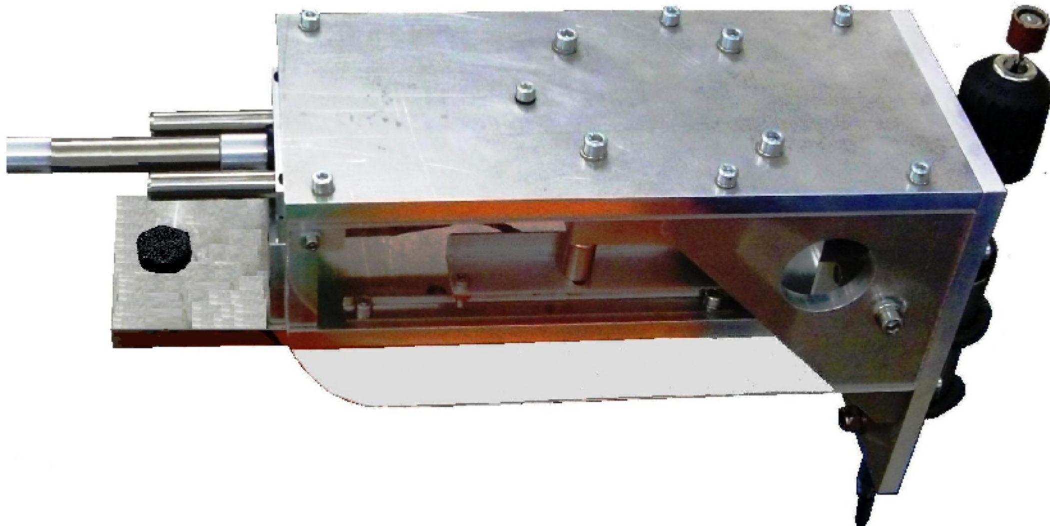


Figura 4.5 – Dispositivo de acabamento construído

Para albergar os componentes pneumáticos necessários ao sistema pneumático de regulação da força, foi mais tarde modelada, construída e aparafusada à base do dispositivo, uma estrutura paralelipipédica em alumínio, com faces em acrílico transparente.

4.3 Sistema pneumático de regulação da força de contacto

A força de contacto, resultante da acomodação das sobreespessuras da peça a maquinar pelo dispositivo de acabamento, é controlada, de uma maneira passiva, com recurso a um sistema pneumático simples.

Fundamentalmente, para além de todos os acessórios de montagem e tratamento do ar, este sistema utiliza um actuador de duplo efeito de baixo atrito e duas válvulas reguladoras de pressão com os respectivos manómetros.

O corpo do cilindro é montado na parte fixa da mesa linear, sendo a sua haste, ligada à corrediça móvel que transporta a estrutura de suporte da ferramenta. O que se pretende, é aplicar uma determinada pressão em cada uma das câmaras do cilindro, de maneira que a força resultante gerada, força essa que corresponderá à força de contacto entre a peça (sapato) e a ferramenta, ocorra sempre no sentido do avanço do actuador (Figura 4.6).

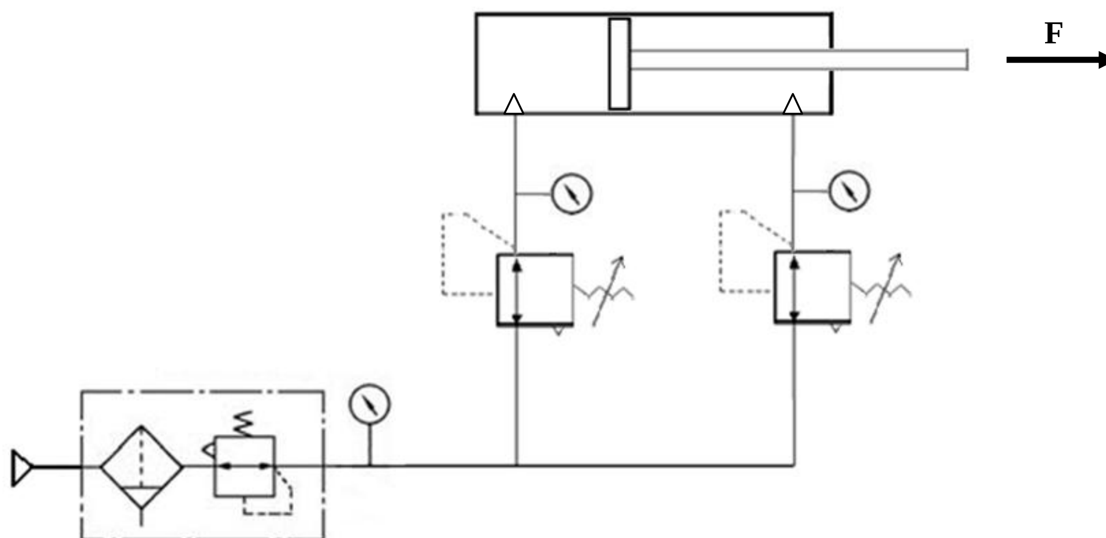


Figura 4.6 – Circuito pneumático de controlo de força

O actuador utilizado para este efeito é o cilindro *MQMLB10*-60D (Figura 4.7), da *SMC*, de duplo efeito, com atrito reduzido e vedação metálica, com 10mm de diâmetro do êmbolo, 4mm de diâmetro da haste e 60mm de curso. Este cilindro possui uma resistência ao deslizamento muito reduzida, sem aumentos da mesma, após períodos de paragem. A sua grande estabilidade permite um controlo de força de apenas 0.05N, dependendo da precisão da regulação da pressão utilizada.



Figura 4.7 – Actuador *SMC MQMLB10*-60D

Apesar de se pretender que a força seja exercida sempre no sentido de avanço do cilindro, optou-se pela utilização de um actuador de duplo efeito, pois a regulação da pressão nas duas câmaras do cilindro tem a vantagem de permitir o ajuste da rigidez do sistema. Esta característica é importante do ponto de vista da versatilidade da aplicação deste sistema a vários processos que envolvam o controlo de força.

Na Tabela 4.2 podem ser consultados os valores teóricos das forças geradas pelo actuador no sentido do avanço, em função dos valores nominais das pressões aplicadas, dentro da sua gama de funcionamento. Estes valores são meramente teóricos, não contemplando factores como o atrito, e foram calculados segundo a expressão:

$$F = P_1 * A_1 - P_2 * A_2 \qquad [S.I.]$$

Na tabela apenas são referidos os valores da força cuja resultante é positiva no sentido do avanço do cilindro.

Força teórica [N]		**P_1 (pressão no avanço) [MPa]**							
		0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
P_2 (pressão no recuo) [MPa]	0	-	7.9	15.7	23.6	31.4	39.3	47.1	55.0
	0.1	-	1.3	9.1	17.0	24.8	32.7	40.5	48.4
	0.2	-	-	2.5	10.4	18.2	26.1	33.9	41.8
	0.3	-	-	-	3.8	11.6	19.5	27.3	35.2
	0.4	-	-	-	-	5.0	12.9	20.7	28.6
	0.5	-	-	-	-	-	6.3	14,1	22.0
	0.6	-	-	-	-	-	-	7.5	15.5
	0.7	-	-	-	-	-	-	0.9	8.8

Tabela 4-2 – Forças nominais em função das pressões aplicadas

As válvulas reguladoras de pressão são válvulas reguladoras de precisão, também da *SMC*, e possuem uma sensibilidade de 0.2%, e uma repetibilidade de +/- 0.5% da pressão aplicada.

Para uma melhor regulação e medição da pressão de funcionamento, utilizaram-se manómetros digitais da série *ISEA30A-01*,também estes da *SMC* (Figura 4.8).



Figura 4.8 – Válvula *SMC IR1020-01* e manómetro *SMC ISEA30A-01*

Estes manómetros têm a particularidade de poder alterar a cor do visor de vermelho para verde ou vice-versa, em função da pressão medida. Ou seja, pode definir-se um valor ou um intervalo de valores para a pressão pretendida, e depois de regular as válvulas para esses valores, sempre que haja uma variação superior ou inferior à pressão desejada, ainda que instantânea e transitória, o visor muda automaticamente de cor.

O tratamento do ar do circuito pneumático de controlo da força é assegurado por um filtro micrónico com um índice nominal de filtragem de 5μm, com regulador e manómetro integrado da *SMC* com a referência *AW20-F01E*.

Verificação da força de acomodação do dispositivo

Para verificar a força de contacto supostamente constante do dispositivo em todo o seu curso de trabalho, foram realizados testes experimentais, nos quais se fez encostar o braço do robô ao mecanismo, fazendo-o deslocar-se segundo o seu grau de liberdade linear na direcção do controlo da força. Na realização destes testes foi utilizado o sensor de força do robô para medir os valores da força durante a execução dos movimentos.

A trajectória criada, representada na Figura 4.9, é composta por quatro movimentos lineares:

1. No primeiro movimento forçou-se o deslocamento do dispositivo desde a sua posição inicial, até próximo do fim de curso (recuo do dispositivo);
2. No segundo movimento fez-se recuar o robô até cerca de metade do curso de trabalho (avanço do dispositivo);
3. No movimento seguinte o dispositivo foi deslocado novamente até próximo do fim de curso (recuo do dispositivo);

4. Finalmente com o último movimento o dispositivo recuperou a posição inicial (avanço do dispositivo).

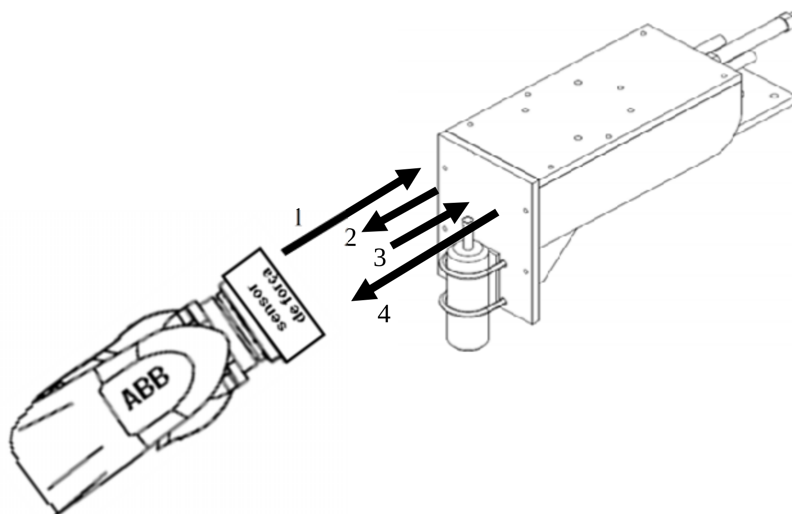


Figura 4.9 – Trajectória utilizada nos ensaios de verificação da força

Nos ensaios efectuados pretendeu-se testar a capacidade de acomodação do dispositivo para diferentes valores de força, e com velocidades diferentes. Assim, as válvulas reguladoras de pressão foram ajustadas para dois valores nominais de força distintos, 9N e 25N, sendo utilizadas duas velocidades, 10mm/s e 50mm/s de execução da trajectória para cada um desses valores de força. Os valores de força foram registados a cada 0.1 segundos.

Nos ensaios efectuados à velocidade de 10mm/s o comportamento do sistema visível na Figura 4.10 apresentou-se idêntico independentemente da força utilizada.

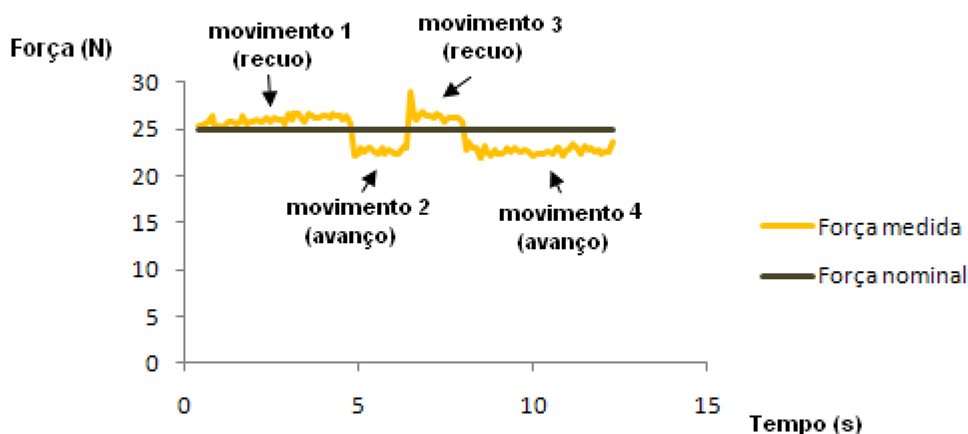


Figura 4.10 – Ensaio de força realizado a 10mm/s com força nominal de 25N

A razão pela qual os valores de força medidos pelo sensor de força apresentam este aspecto, acima da força nominal no movimento de recuo, e abaixo desta durante o avanço, prende-se com o facto do sentido da força de atrito do mecanismo sofrer uma inversão durante a execução da trajectória (Figura 4.11)

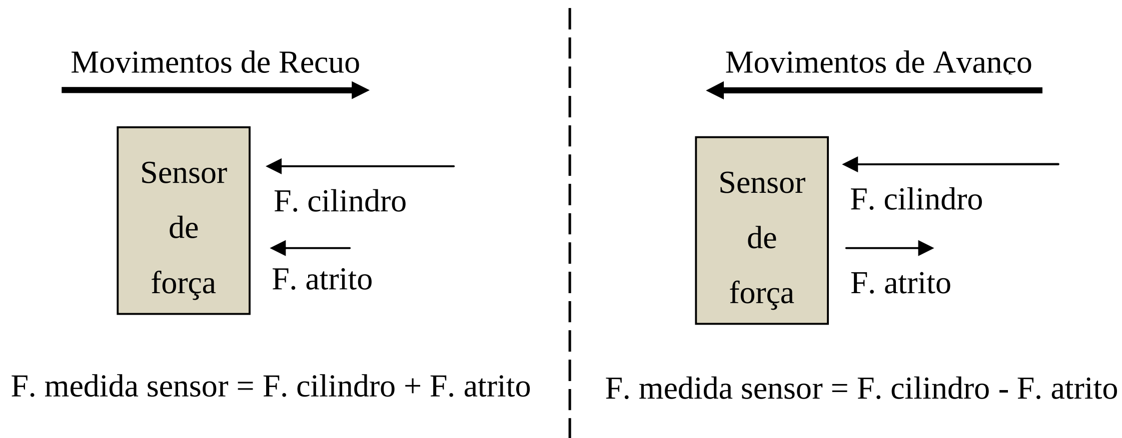


Figura 4.11 – Forças envolvidas no sistema

Esta força de atrito é composta pelo atrito existente no sistema de guiamento da mesa linear, o atrito entre o êmbolo e o corpo do cilindro pneumático, e o atrito viscoso do ar, que neste caso representa um valor entre 2 a 3 Newton.

Nos ensaios realizados à velocidade de 50mm/s, o sistema apresenta um comportamento dinâmico pior, devido à velocidade superior utilizada (Figura 4.12).

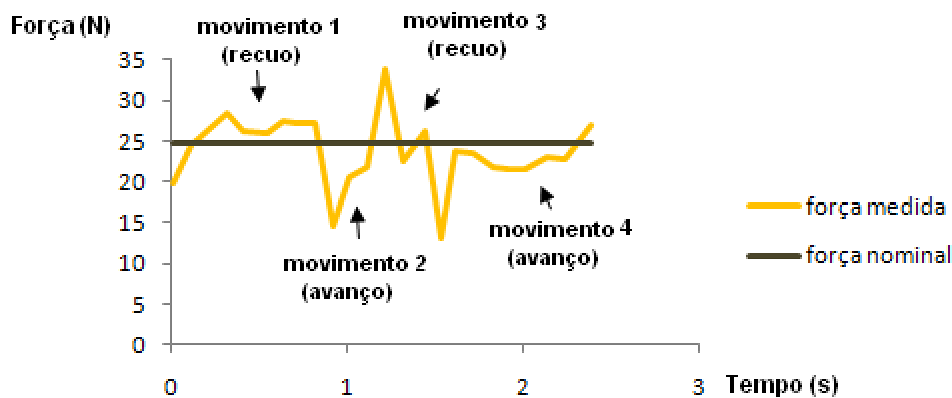


Figura 4.12 – Ensaio de força realizado a 50mm/s com força nominal de 25N

Pode claramente observar-se que a força apresenta “picos” com valores máximos e mínimos sempre que o sentido do movimento da trajectória é invertido. Este comportamento pode ser explicado pela natureza construtiva do dispositivo. Uma vez que o controlo de força

é realizado de maneira passiva, nestes valores está certamente presente uma forte componente inercial da massa dos componentes móveis do mecanismo, que se traduz na influência da força medida pelo sensor durante as acelerações/desacelerações ocorridas nas inversões do sentido linear de movimento.

Neste ensaio em questão, em que foi utilizada uma velocidade bastante elevada, o facto de os valores de força terem sido registados apenas a cada intervalo de tempo de 0.1 segundos, poderá também ter contribuído para este aspecto mais “quebrado” da linha da força ao longo do tempo.

4.4 Sistema electrónico de medição, monitorização e comunicação

Para detectar e medir o deslocamento provocado pela acomodação do dispositivo à peça a maquinar, e enviá-lo através da porta série para o controlador do robô, foi construído um sistema electrónico de medição, monitorização e comunicação da posição da ferramenta.

Como componentes principais, este sistema possui um micro controlador, dois reguladores electrónicos de tensão, um *LCD*, um conector RS-232, e a ligação para o codificador incremental.

Alimentação

Uma vez que todos os componentes introduzidos neste circuito electrónico funcionam com um valor de tensão nominal de 5V, uma solução simples, barata, eficaz, e bastante prática para a alimentação deste sistema, foi a utilização de um vulgar carregador recuperado do conhecido fabricante de telemóveis *NOKIA*, o modelo *ACP-12E*. O carregador em questão tem como valores nominais de saída de 5,7V para a tensão e 800mA para a corrente.

O valor desejado e uniforme de tensão de 5V para a alimentação de todos os componentes do sistema electrónico, foi obtido com a utilização de um regulador de tensão programável *MAX 667*, realizada uma montagem de acordo como mostrado na Figura 4.13.

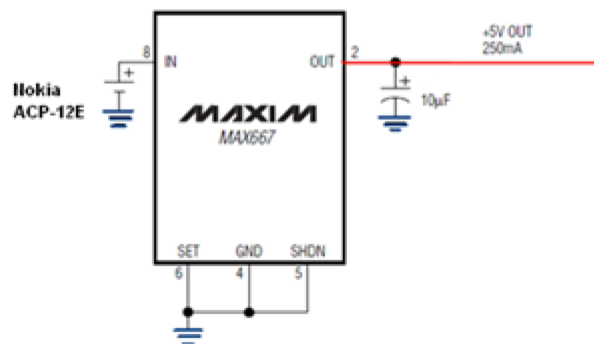


Figura 4.13 – Diagrama de ligações do regulador de tensão

Codificador incremental

O *encoder* incremental utilizado no dispositivo auxiliar de acabamento provem de uma impressora *HP Deskjet 690C*. Tanto a cabeça de leitura como a fita se encontravam em bom estado, pelo que foi assim possível incorporá-los no dispositivo auxiliar de acabamento.

A fita utilizada tem uma resolução de 150LPI (linhas por polegada). Como o sinal medido está em quadratura, com este *encoder* digital consegue obter-se uma resolução na medição do deslocamento de 0.042mm (25.4/600).

Ou seja, a posição da ferramenta é medida em impulsos, gerados pelo movimento relativo da fita em relação ao *encoder*, sendo cada impulso correspondente a um deslocamento de aproximadamente 0,042mm.

Micro controlador

O micro controlador utilizado é o *PIC 18F2431*, e é o elemento central de controlo deste dispositivo (Figura 4.14). Este micro controlador da *MicroChip* foi escolhido para integrar este circuito, por pertencer a uma família de micro controladores cujas características se adaptam perfeitamente aos requisitos impostos.

Este componente possui uma interface de comunicações série, *SCI*, denominada *EUSART* (*Enhanced Universal Synchronous Asynchronous Receiver Transmitter*), que permite realizar a comunicação série com outros dispositivos, e uma interface para ligação do codificador incremental *QEI* (*Quadrature Encoder Interface*) com capacidade para medir velocidades, detectar sentidos e mudanças de sentido. Possui também um oscilador interno que pode ser programado para a frequência de relógio desejada, eliminando assim a

necessidade de incorporar cristais externos no sistema electrónico. A frequência utilizada foi 8MHz.

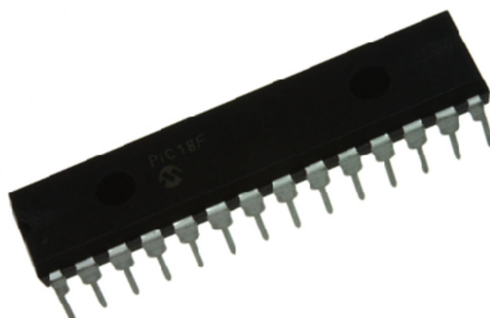


Figura 4.14 – Micro controlador PIC 18F2431

Estas características a par de entre muitas outras que este elemento possui, conferem um grande potencial a este micro controlador, associado a uma elevada flexibilidade com uma vasta gama de aplicações.

A programação deste elemento foi feita com recurso à placa electrónica de desenvolvimento *EasyPic 4* da *mikroElektronika*, e o respectivo *software* de programação *mikroC*. O código da programação utilizada neste sistema encontra-se no Anexo B.

Comunicação

A comunicação dos dados de posição do sistema electrónico desenvolvido, com o robô, é realizada através da comunicação série, um tipo de comunicação *ponto-a-ponto*.

Existem fundamentalmente dois tipos fundamentais de comunicações *ponto-a-ponto*, a comunicação paralela e a comunicação série. A diferença entre elas tem a ver com o canal de comunicação, ou seja, o recurso físico que conduz a informação desde um emissor a um receptor.

A comunicação paralela (Figura 4.15), conta com um canal de comunicação que permite transmitir vários *bit*s em simultâneo, tipicamente 8 *bits* (em paralelo).

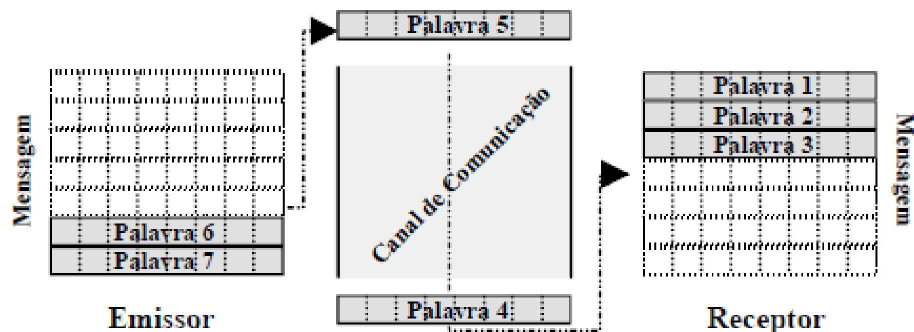


Figura 4.15 - Comunicação paralela [15]

Na comunicação série (Figura 4.16), o canal de comunicação apenas consegue transmitir um *bit* de cada vez. Por isso, os *bits* que compõem uma mensagem são, neste caso, transmitidos sequencialmente, ou em série [Magalhães, 2001].

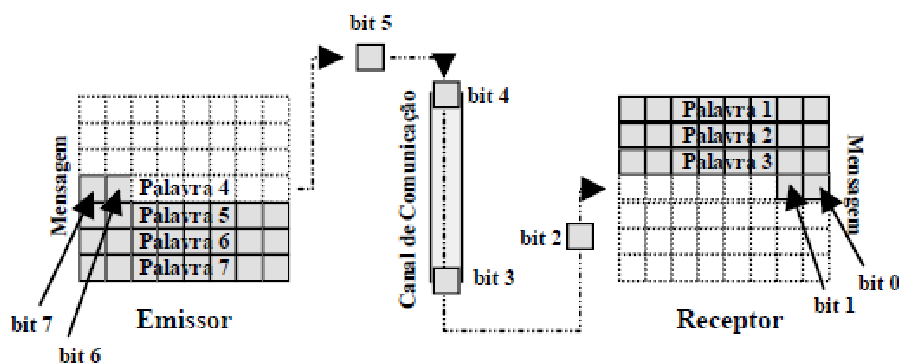


Figura 4.16 – Comunicação série [15]

Estes dois tipos de comunicações apresentam vantagens e desvantagens, a nível de rapidez, ruído, e implementação relativamente à maior ou menor complexidade dos protocolos utilizados. Fundamentalmente, a comunicação paralela apresenta a grande vantagem de ser mais rápida que a comunicação série, contudo tem uma baixa imunidade ao ruído, sendo aplicada em pequenas distâncias. A comunicação série tem a particularidade de ser aplicável a distâncias maiores dada a sua elevada imunidade ao ruído, é mais económica e mais difundida dado que praticamente todos os sistemas operativos a suportam através da norma RS-232-C, normalmente referida apenas por RS-232.

Não menos relevante é o facto da generalidade dos sistemas industriais programáveis utilizarem precisamente esta interface para a descarga de programas desenvolvidos

---

[15] Imagem retirada dos acetatos de apoio às aulas da disciplina Computação Industrial de 2001 da FEUP.

normalmente em computadores pessoais. É o caso dos controladores lógicos programáveis, controladores de temperatura, controladores de motores, e no caso presente o controlador *IRC5* do Robô.

Comunicação RS-232

Pelas razões anteriormente apresentadas, a comunicação série foi o tipo de comunicação escolhido para fazer a ligação entre o sistema electrónico desenvolvido e o robô.

Existem diversas normas relativas à representação de um “1” e de um “0” no contexto da comunicação série. A norma RS-232 define que, do ponto de vista do emissor, o nível lógico “0” corresponde à aplicação de uma tensão entre 5 e 15 Volt. Também do ponto de vista do emissor, o nível lógico “1” corresponde a uma tensão entre -15 e -5 Volt. Já do ponto de vista do receptor, a norma RS-232 define os níveis lógicos “0” e “1” como correspondendo aos intervalos [3, 15] e [-15, -3] Volt, respectivamente (Figura 4.17). Assim, uma comunicação estabelecida no âmbito desta norma nunca tem uma margem de ruído inferior a 2V [Magalhães, 2001].

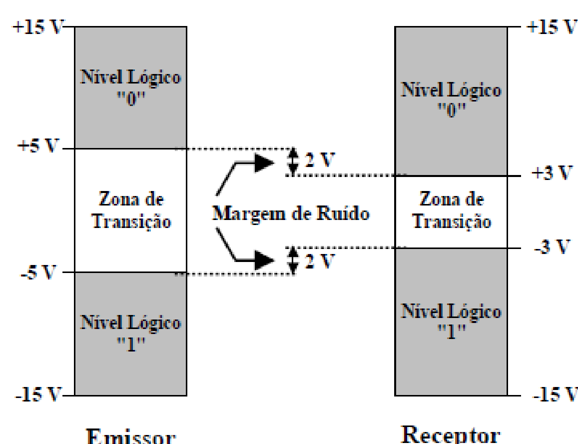


Figura 4.17 – Gama de tensões para a comunicação RS-232 [16]

Uma vez que esta norma utiliza um intervalo de tensão entre -15 e 15V, e o micro controlador (e os restantes componentes do sistema electrónico) são alimentados com uma tensão nominal de 5V torna-se imperativo integrar um conversor de níveis de tensão, neste tipo de circuitos. O componente utilizado foi o *MAX232* da *MAXIM* (Figura 4.18).

[16] Imagem retirada dos acetatos de apoio às aulas da disciplina Computação Industrial de 2001 de FEUP.



Figura 4.18 – MAX 232

Com o uso deste componente os valores de tensão correspondentes aos níveis lógicos "0" e "1" ao nível da emissão do sinal, são respectivamente 10V e -10V. Basicamente este conversor de tensão permite duplicar o valor de tensão aplicada e inverter o seu sinal caso seja necessário (ver diagrama de funcionamento apresentado na Figura 4.19).

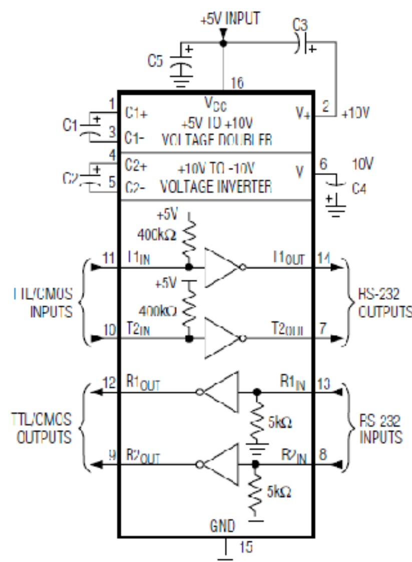


Figura 4.19 – Diagrama de funcionamento do *MAX232*

Codificação da informação

A informação da posição medida em impulsos pelo codificador incremental, que é enviada através da porta série, está contida numa palavra composta por 2 *bytes*.

São necessários 2 *bytes* para conter a informação da posição da ferramenta, pois apesar do curso máximo da ferramenta rondar os 50mm, dada a resolução do *encoder*, a posição pode facilmente corresponder a um número inteiro maior que 255. Este valor, é o máximo valor que se obteria caso fosse utilizado um só *byte* para transportar a informação da posição.

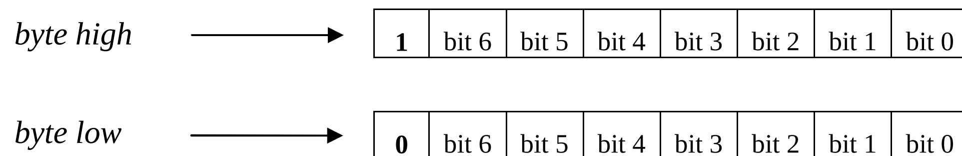
1 *byte* = 8 *bits*

$$2^8 - 1(\text{valor } 0) = 255$$

Assim, um dos *bytes* que contém a informação da posição será o *byte high* e o outro, o *byte low*, contendo este último a parte menos significativa do número inteiro em questão.

Sabendo que posteriormente esses *bytes* terão que ser lidos e interpretados pelo controlador do robô, é necessário assegurar que o controlador os identificará correctamente, para o seu tratamento posterior.

Uma primeira abordagem previa enviar sequencialmente 3 *bytes* de cada vez, sendo enviado um *byte* inicial correspondente a um valor conhecido, antes dos *bytes high* e *low*. Contudo concluiu-se que o processo de leitura e processamento da informação por parte do controlador do robô, se tornaria mais expedito se os *bytes high* e *low* fossem codificados, tirando partido do facto de se realizar estas operações ao nível da programação do micro controlador. Assim, através de operações binárias, codificaram-se os 2 *bytes* atribuindo o valor 1 (um), ao *bit* mais significativo do *byte high*, e 0 (zero), ao *bit* mais significativo do *byte low*, estando a informação da posição contida nos restantes 14 *bits* da palavra.



A codificação desta informação foi feita ao nível do micro controlador, aquando da sua programação e é descrita em detalhe no Anexo C.

Botão de reset

Na construção deste sistema foi necessário introduzir um botão mono estável que, ao ser premido, reinicia o contador do módulo do codificador incremental com um valor de referência correspondente ao deslocamento nulo do sistema de controlo de força que transporta a ferramenta.

Uma vez que a informação da posição está contida em 2 *bytes*, e o sistema apresenta uma elevada resolução e sensibilidade, foi arbitrado o valor 10 para a posição de referência.

Caso fosse utilizado o valor 0 (zero) para esta referência, o sistema, ao regressar à posição inicial após um qualquer deslocamento, poderia ter tendência a readoptar a posição inicial com um valor inferior. Como na escala por nós adoptada o valor imediatamente inferior a 0 (zero), não é -1 mas sim 16383 ($16383=2^{14} – 1$), isto poderia introduzir-nos dados erróneos na leitura da posição e problemas mais tarde ao nível do controlo da velocidade da trajectória do robô.

Assim, o intervalo [0,10] funciona como um intervalo de segurança para a posição de referência.

Visualização da posição

Na construção da placa, foi incorporado também um LCD para indicação visual, e em tempo real do deslocamento medido pelo codificador incremental, e enviado via RS-232 para o robô. O LCD utilizado tem duas linhas, cada uma com dezasseis caracteres, sendo cada um destes, constituído por quarenta pixéis (Figura 4.20). Foi utilizada uma montagem de 4*bits* com o micro controlador, que utiliza somente quatro linhas de dados.

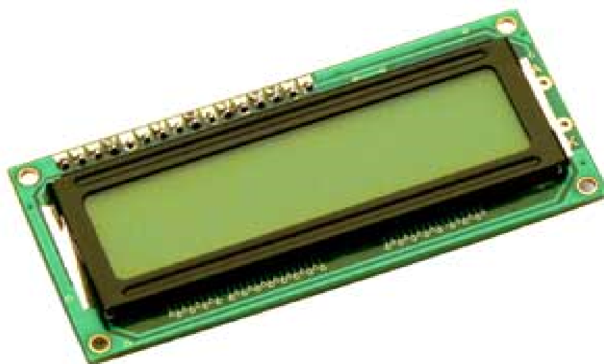


Figura 4.20 - LCD

O LCD é utilizado para se obter uma indicação visual do valor da posição medido pelo *encoder*, e transmitido através da porta série para o controlador do robô.

A imagem da Figura 4.21 apresenta um diagrama esquemático do circuito electrónico, e das ligações criadas entre os principais componentes.

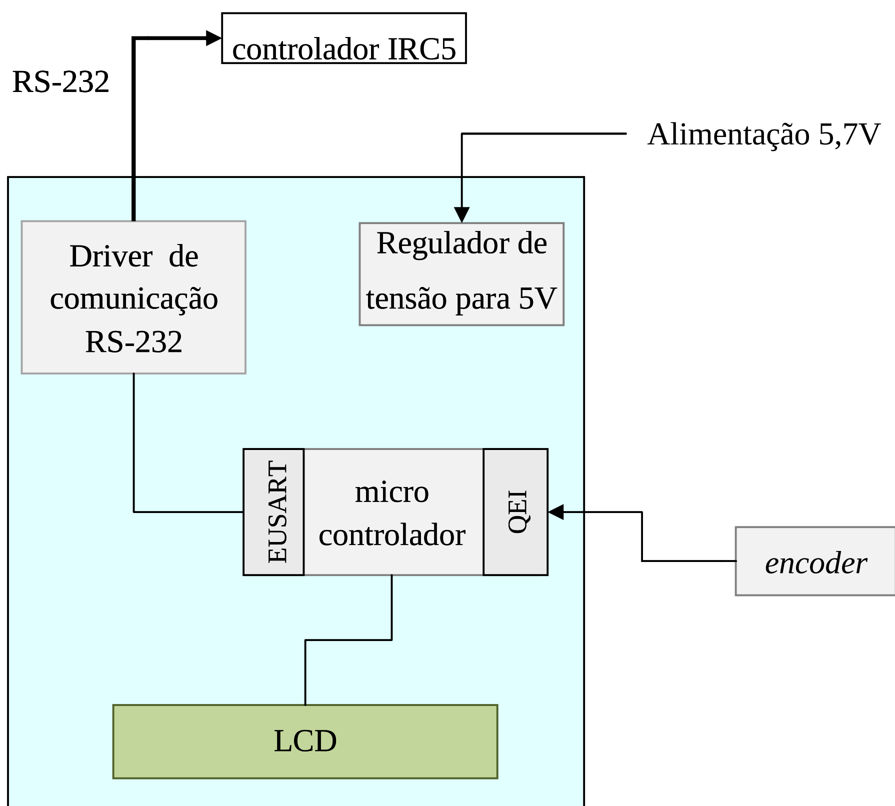


Figura 4.21 – Esquema dos componentes do sistema electrónico

As imagens da Figura 4.22, a par da Tabela 4.3, apresentam o aspecto final do sistema electrónico desenvolvido e referenciam/localizam os principais componentes utilizados.

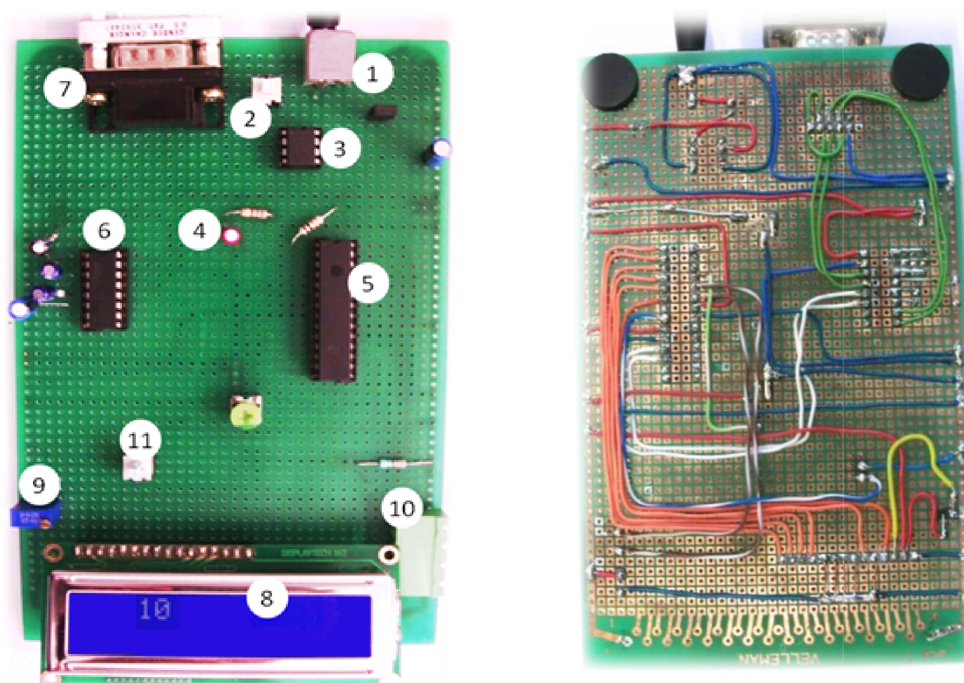


Figura 4.22 – Sistema electrónico desenvolvido

Nº	Componente
1.	Ficha de alimentação
2.	Interruptor "*on/off*"
3.	Regulador de tensão MAX 667
4.	Led indicador de funcionamento
5.	PIC 18F2431
6.	Conversor de níveis de tensão MAX 232
7.	Ficha RS-232
8.	LCD
9.	Potenciómetro de ajuste do contraste do LCD
10.	Ficha de ligação ao codificador incremental
11.	Botão de reset

Tabela 4-3 – Componentes do sistema electrónico

Verificação da dinâmica de comunicação entre o dispositivo e o controlador *IRC5*

Para avaliar a dinâmica da comunicação entre o sistema electrónico desenvolvido e o controlador do robô, foi executado um teste de leitura e tratamento de valores de posição por parte do controlador do robô.

Para isso foi desenvolvido um programa para o controlador do robô. Este programa utiliza uma rotina intitulada *GetData*, desenvolvida para o efeito, que adquire, identifica e reconstrói os *bytes* que contém a informação da posição. O programa é executado durante 100 vezes, registando-se o tempo que demora a ler a porta série e a processar a informação. O tempo de execução deste teste foi de 1.063 segundos.

Idealmente o tempo nunca poderia ser inferior a 1 segundo pois na programação do micro controlador introduziu-se um atraso de 10 milissegundos, entre cada conjunto de dois *bytes* (*high* e *low)*, que compõe a informação da posição. Este resultado demonstrou uma boa dinâmica de comunicação entre o dispositivo e o controlador do robô.

4.5 Conclusão

Mecanicamente este dispositivo de controlo de força desenvolvido possui um grau de liberdade linear, segundo o qual a ferramenta se desloca. Tem um curso máximo de trabalho de 52mm regulável, uma capacidade de regulação da força de contacto que pode atingir valores até 50N. À excepção dos varões de guiamento da mesa que são de aço inoxidável, todos os restantes componentes estruturais são em alumínio. A sua massa total ronda os 6,5kg.

O dispositivo possui um sistema electrónico de medição da posição da ferramenta com uma resolução de 0,042mm, e comunicação de dados através da porta série, utilizando os princípios da comunicação RS-232.

Na Figura 4.23 é apresentado o dispositivo completo, onde pode ser visto o sistema electrónico de medição e comunicação desenvolvido, o sistema pneumático de regulação da força e o sistema de posicionameno do suporte da ferramenta.

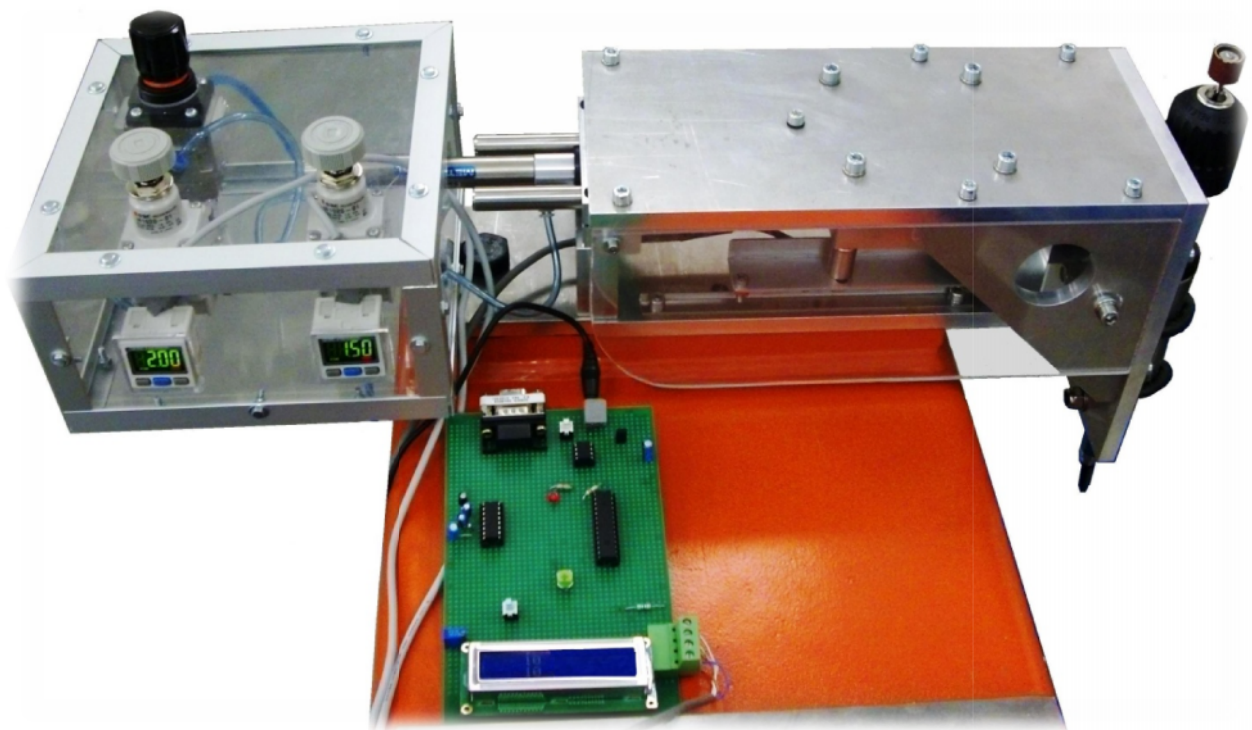


Figura 4.23 – Dispositivo de controlo de força desenvolvido

5 . Implementação do Sistema Desenvolvido

5.1 Introdução

Este quinto capítulo está centrado na integração funcional dos componentes desenvolvidos na célula robotizada e na sua programação. Todos os procedimentos efectuados para testar e validar a solução proposta são aqui detalhadamente explicados, bem como os ensaios efectuados e os respectivos resultados experimentais.

5.2 Programação o*ff-line* da célula robotizada

Após a montagem e configuração do dispositivo de acabamento na mesa do robô e verificação de funcionamento do sistema electrónico, foi realizada a programação da célula robotizada.

Actualmente existem dois métodos distintos para a programação de robôs. A programação *online* e a programação *off-line*.

A programação *online*, como o nome sugere, é realizada com a utilização directa do robô. As trajectórias são ensinadas através do guiamento do robô até aos pontos (ou *targets*), pretendidos que são memorizados, sendo a sua trajectória construída a partir destes. O guiamento do robô pode ser executado manualmente, com o operador a pegar no braço do manipulador e a levá-lo ao ponto desejado, ou através da consola de programação fazendo para isso uso de um *joystick.*

Apesar de este método ser particularmente simples de implementar, tem a desvantagem de ser necessário interromper a laboração da célula robotizada para a sua realização, além de aumentar a probabilidade de ocorrerem acidentes uma vez que o operador está dentro do espaço de trabalho do robô.

A programação o*ff-line,* como o nome indica, é feita sem recurso à utilização directa do robô. Serve-se da utilização de *software* de programação adequado, com uma interface gráfica interactiva, onde é possível incorporar todos os elementos de uma célula robotizada a partir de uma base de dados do respectivo fabricante, ou até criar ou importar novos componentes como ferramentas e peças anteriormente modeladas noutro tipo de *software*.

Este método tem a vantagem de não ser necessária a interrupção do trabalho na célula robotizada, bem como de antecipar potenciais erros de programação ou colisões entre os elementos da célula, através da visualização das simulações cinemáticas criadas.

O *RobotStudio*

O *software* de programação utilizado foi o *RobotStudio* da *ABB Robotics*, juntamente com a aplicação específica para operações de maquinagem, o *Machining Powerpack*.

O *RobotStudio* apresenta inúmeras características que permitem ao utilizador:

- Importar geometrias modeladas em programas de *CAD*, que podem ser mais tarde definidas como peças, ferramentas, ou outros componentes da célula;
- Gerar trajectórias automaticamente a partir de um modelo *CAD*;
- Detectar colisões e verificar o alcance do robô;
- Utilizar aplicações de *software* específicas para operações que envolvam maquinagem, soldadura, entre outras;
- Verificar, através de simulação gráfica, as trajectórias programadas;
- Verificar a aplicabilidade do programa na célula real;
- Introduzir imediatamente um programa desenvolvido no controlador real do robô;
- Criar células robotizadas com controladores virtuais a partir de cópias do controlador real;
- Programar o robô virtual de uma maneira idêntica à programação *on-line*, uma vez que possui um modelo virtual interactivo da consola real.

A Figura 5.1 mostra a interface gráfica da célula robotizada virtual utilizada na programação *off-line* da célula real.

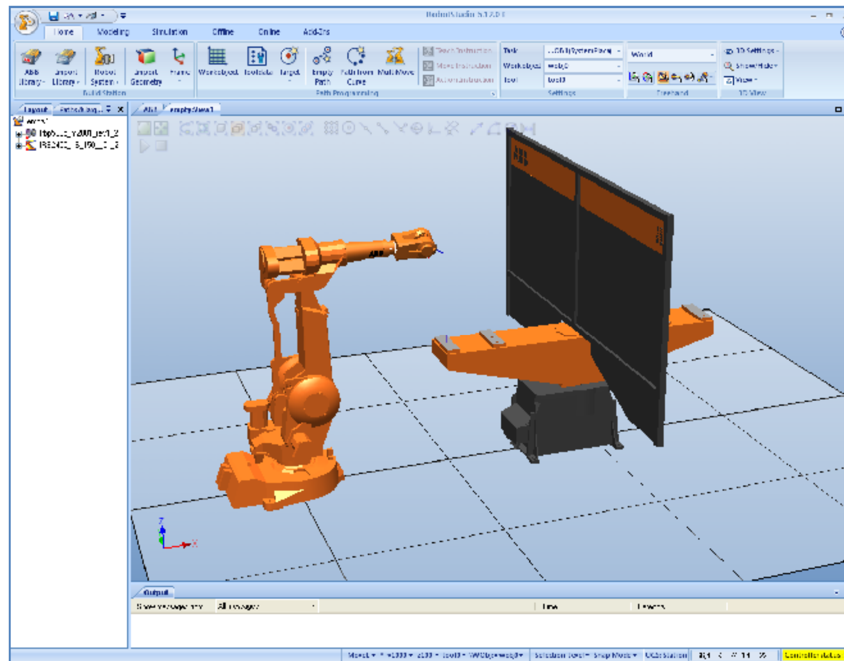


Figura 5.1 – Interface gráfica do *RobotStudio*

Os controladores aqui referidos, quer o virtual, quer o real, não são mais do que postos de processamento de dados capazes de interpretar informações inseridas numa linguagem própria, onde a sua interpretação se traduz na realização de acções por parte do robô. Assim, a partir de uma cópia do controlador real, criou-se ao nível do *software* um controlador virtual. Isto possibilita que haja uma concordância entre o modelo virtual e o real, sendo possível a transferência imediata do programa para a célula minimizando os procedimentos de operações de calibração como, por exemplo, o reposicionamento dos referenciais de trabalho ou da ferramenta [Moita, 2009].

Simulação cinemática do processo de acabamento

A simulação cinemática do processo de acabamento em ambiente virtual, iniciou-se com a importação das geometrias do dispositivo de acabamento e de um sapato, previamente modelados em *software* adequado, com o seu correcto posicionamento na célula.

O sapato modelado é constituído por uma sola dupla, o corpo do sapato propriamente dito e o seu suporte (Figura 5.2).



Figura 5.2 – Sapato modelado [17]

Como a Figura 5.3 evidencia, o dispositivo foi fixo e está associado à mesa rotativa, enquanto o sapato está associado ao robô.

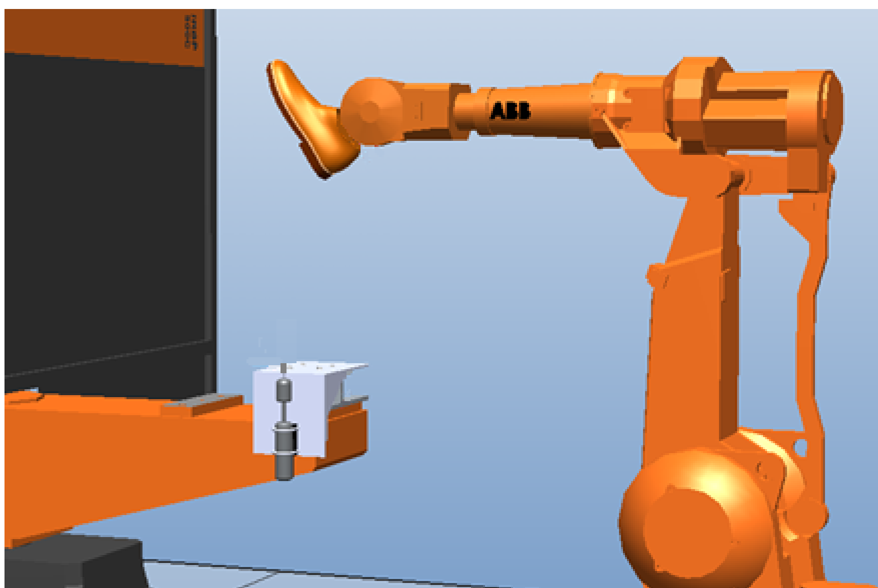


Figura 5.3 – Implementação da célula robotizada virtual no *Robotstudio*

Geração da trajectória

Em robótica, uma trajectória é um percurso realizado pelo manipulador, construído com um conjunto de instruções de movimento entre pontos. Essas instruções de movimento, para além da definição das coordenadas dos pontos e da velocidade de execução do movimento, contém ainda outras informações implícitas, sobre a ferramenta utilizada, o referencial de trabalho da peça, e a precisão posicional com que o robô executa o movimento.

[17] Sapato modelado em *SolidWorks* por Nuno Moita, para o projecto de dissertação do próprio intitulado "*Concepção e Simulação de uma Célula Robotizada para Acabamentos de Solas de Sapatos*".

A trajectória que se pretende realizar consiste na movimentação do sapato ao longo de todo o contorno da sua sola, assegurando um contacto sempre perpendicular entre esta e a direcção linear do grau de liberdade do dispositivo de acabamento.

De referir que na simulação virtual, o dispositivo de acabamento tem um comportamento rígido, servindo apenas esta simulação para avaliar a exequibilidade da operação do ponto de vista cinemático e, mais tarde, utilizar e modificar o código de programação na linguagem *RAPID* resultante da sincronização para o controlador virtual.

A linguagem de programação *RAPID* é a linguagem utilizada pela *ABB* para a programação dos seus robôs. A alteração posterior desse código visa incluir instruções para a leitura da porta série e, com base no tratamento dessa informação, a subsequente actualização da velocidade de execução da trajectória.

Existem várias maneiras de definir a trajectória a executar. Ela pode ser definida manualmente, ou através de um procedimento mais expedito praticamente automático.

Decidiu-se então que a trajectória seria gerada segundo o contorno da sola, definido pela superfície de interface entre a sola superior e a sola inferior do sapato modelado.

Se a trajectória for gerada manualmente, numa primeira fase, é necessário definir uma curva do tipo "*spline*" e, numa segunda fase, o contorno delimitado pelas duas solas em cima da qual serão criados os pontos utilizados para as instruções de movimento do robô. Esses pontos podem ser criados com um afastamento mínimo predefinido entre eles, sendo fulcral definir uma orientação correcta dos referenciais dos mesmos, de modo a assegurar que a sola contactará com a ferramenta na orientação pretendida.

Posteriormente será ainda conveniente criar pontos que definam a aproximação e o afastamento da peça à ferramenta, respectivamente, antes e depois da operação de maquinagem propriamente dita.

Todos esses pontos serão incorporados numa trajectória, tornando-se ainda necessário definir o tipo de instruções de movimento, a configuração adoptada para o robô, a velocidade da trajectória, a precisão posicional entre outros parâmetros.

O Machining PowerPack

A utilização desta aplicação do *RobotStudio* torna o processo de geração de trajectórias para maquinagem mais hábil e insensível a erros de configuração. Descreve-se então de seguida o processo de geração de uma trajectória aplicando este método.

Uma vez importados e correctamente posicionados na célula virtual, o sapato e o dispositivo de acabamento, torna-se necessário definir o referencial da ferramenta, o *TCP* (*tool center point)*, fazendo-o coincidir com o eixo desta, e impor a condição de que se encontra fixa, e não transportada pelo robô (Figura 5.4).

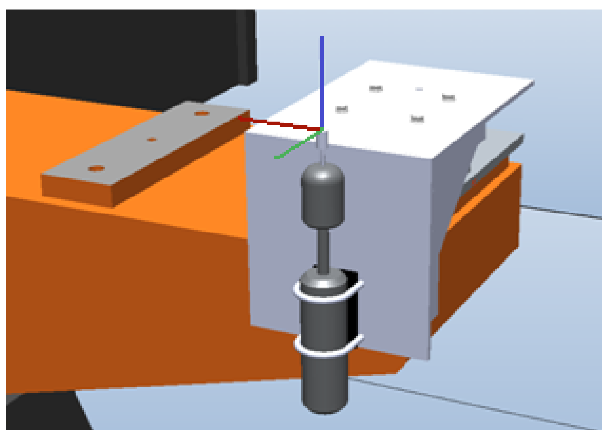


Figura 5.4 – Atribuição do referencial da ferramenta

Seguidamente, através da execução do assistente de soluções *"solution wizard"*, a configuração da solução é alcançada, sendo para isso necessário definir determinados parâmetros ao longo de sete etapas, que serão aqui brevemente descritas.

1. Definição da solução de maquinagem

Nesta primeira etapa, é definida genericamente a solução de maquinagem, com base na escolha do controlador do robô que vai executar a operação de maquinagem, da respectiva tarefa onde se insere essa operação e de um nome para o processo (Figura 5.5).

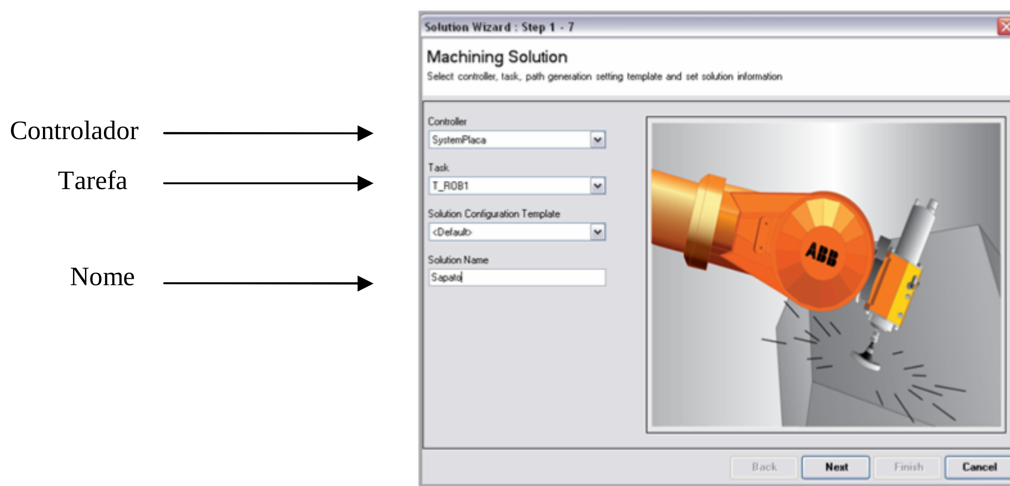


Figura 5.5 – Definição da solução de maquinagem

2. Definição das superfícies de maquinagem

Nesta fase são seleccionadas as superfícies que vão ser maquinadas, sendo neste caso as superfícies laterais da sola do sapato (Figura 5.6).

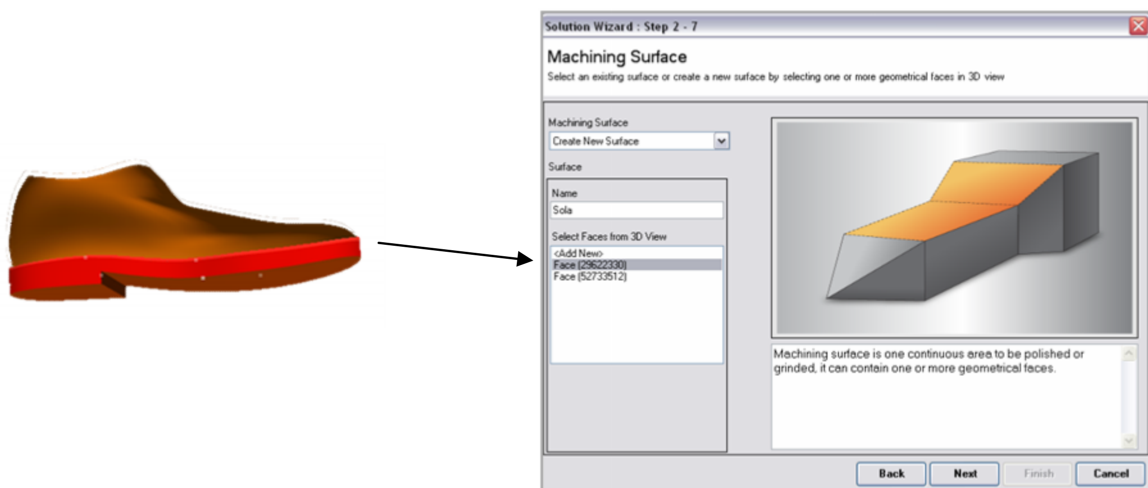


Figura 5.6 – Definição das superfícies de maquinagem

3. Definição dos parâmetros do processo de maquinagem

No terceiro passo, são definidos alguns parâmetros do processo de maquinagem, como a tolerância máxima, o ângulo de desvio de trajectória, o tipo de processo de maquinagem, a possibilidade de optimização da velocidade na superfície da peça, entre outros (Figura 5.7).

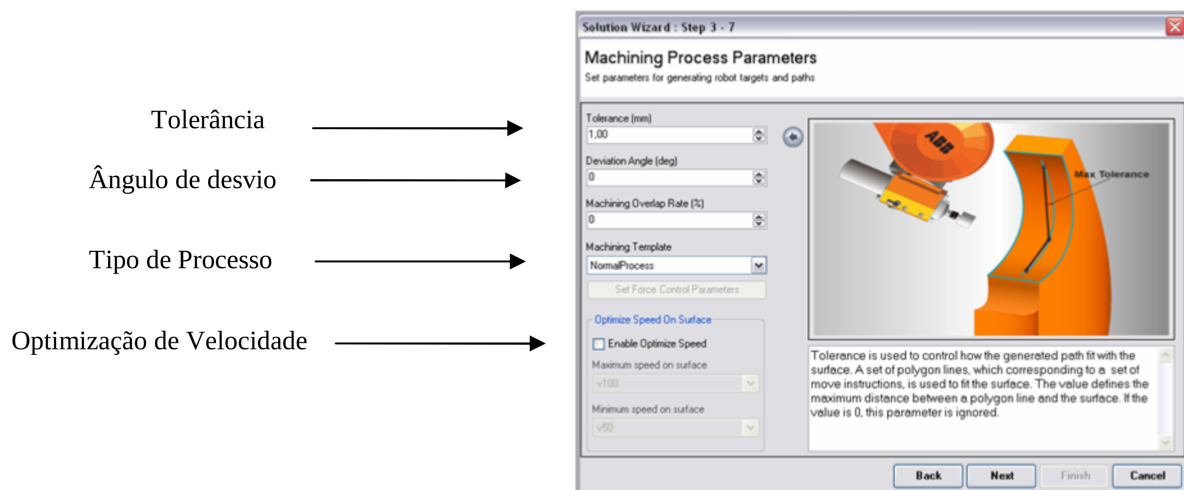


Figura 5.7 – Definição dos parâmetros de maquinagem

4. Definição da ferramenta e dos referenciais de trabalho

Nesta fase é escolhido o referencial de trabalho associado à peça e a ferramenta pretendida. São também aqui definidas as características da ferramenta, como o seu comprimento, raio e o modo de trabalho desta (Figura 5.8).

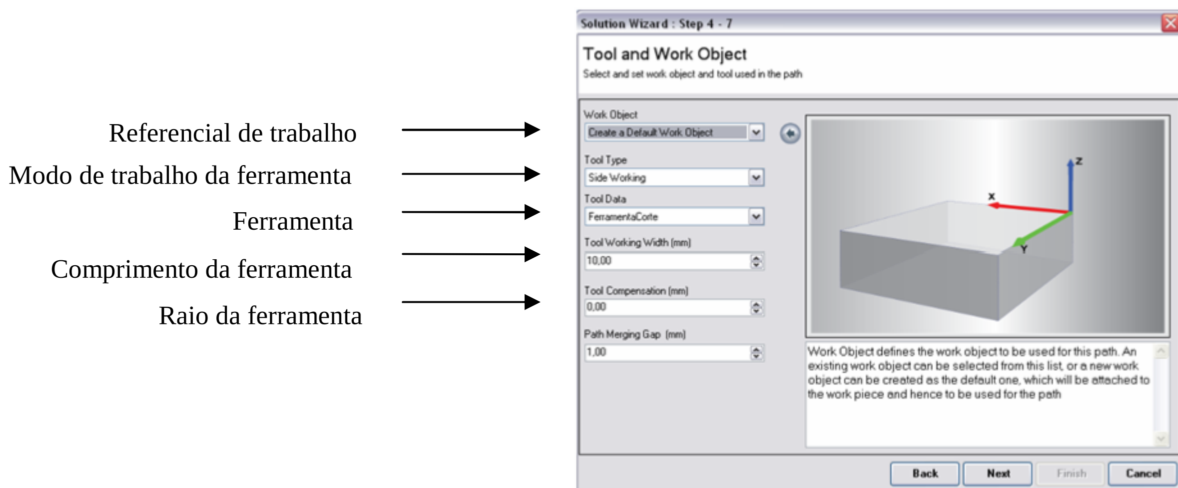


Figura 5.8 – Definição da ferramenta e dos referenciais de trabalho

5. Tipo de geração da trajectória

Nesta etapa é definido o tipo de geração de trajectória pretendido. Neste caso, o tipo de trajectória gerada que melhor se adequa a este processo, é uma curva coincidente com a aresta que envolve todo o contorno da sola do sapato, localizada na interface entre a sola superior e a sola inferior (linha a azul na Figura 5.9).

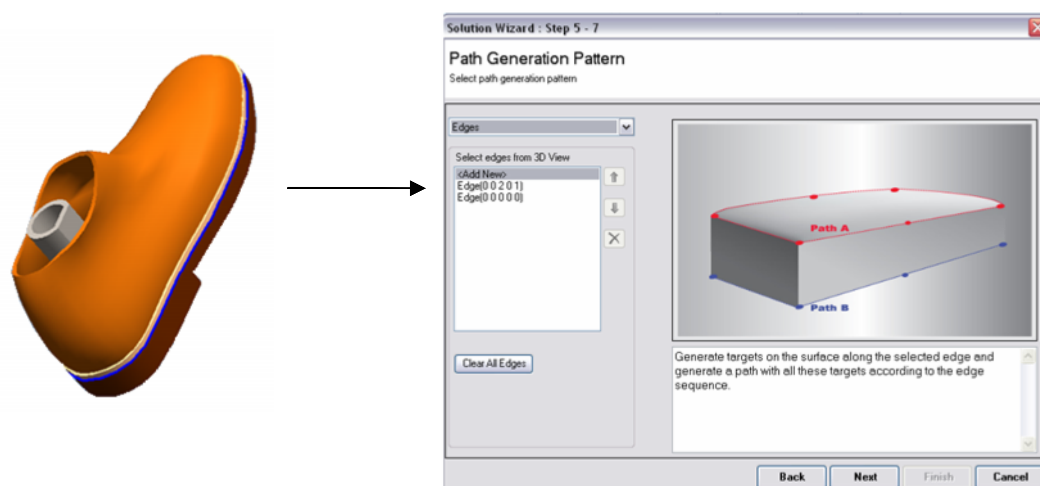


Figura 5.9 – Definição da geração da trajectória

6. Definição da orientação da trajectória

Aqui, entre outros parâmetros, é definida a orientação dos pontos em relação à trajectória, o ângulo de trabalho da ferramenta, o tipo de ligação entre trajectórias, e trajectórias de aproximação e afastamento (Figura 5.10).

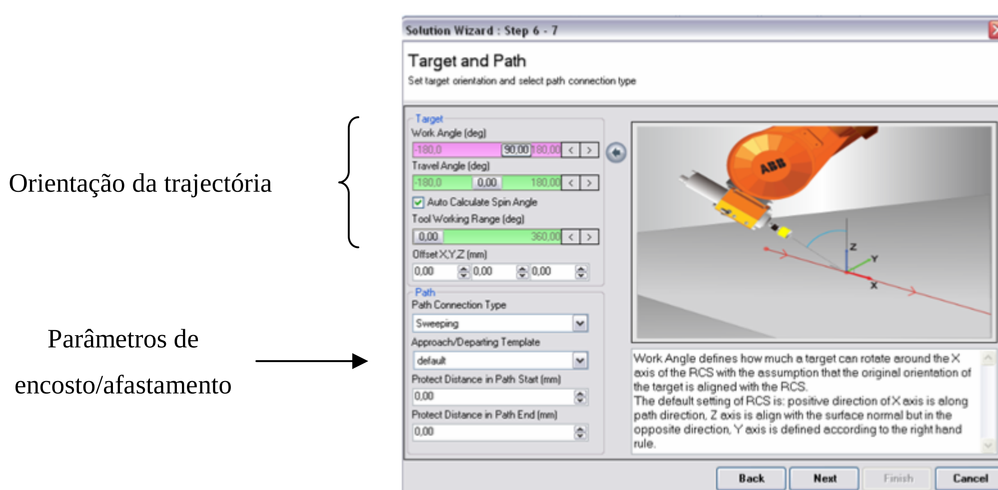


Figura 5.10 – Definição da orientação dos pontos da trajectória

7. Pré visualização da trajectória

Neste último passo é possível ver a trajectória gerada, bem como o seu sentido. É também possível atribuir uma configuração automática dos pontos da trajectória para robô, e fazer a sua sincronização para o controlador virtual (Figura 5.11).

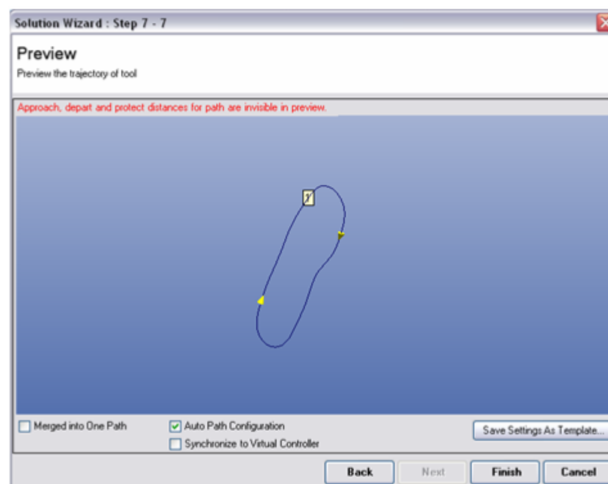


Figura 5.11 – Pré-visualização da trajectória criada

Edição do código de programação *RAPID*

Tendo obtido uma solução cinemática virtual, e feita a sua sincronização para o controlador virtual do *RobotStudio*, é obtido o programa da simulação em termos de código *RAPID* de programação do robô.

Este código foi editado, a fim de se incorporarem instruções para a leitura dos valores de posição da ferramenta enviados através da porta série, sua descodificação e instruções para a actualização da velocidade.

Leitura da porta série

A correcta aquisição dos valores de posição da ferramenta, enviados através da porta série, condiciona todo o processo de actualização de velocidade. Assim, o código obtido, resultante da simulação cinemática virtual, foi editado a fim de incorporar instruções que possibilitam a leitura, identificação e reconstrução da informação da posição.

Estes procedimentos são explicados em pormenor no Anexo E, bem como as instruções de programação que foram adicionadas para tal.

Actualização da velocidade

Após uma análise das instruções de programação do código *RAPID*, identificaram-se duas possibilidades para se fazer a actualização do valor da velocidade do robô durante a execução de um programa. Essas instruções são o *SpeedRefresh* e o *Velset*. Em qualquer dos casos, o cálculo do novo valor de velocidade ordenado ao robô é feito dentro da rotina *GetData*, com base na informação da posição. Esta rotina foi programada para processar a informação da posição da ferramenta disponibilizada via porta série.

- ***VelSet***

O *VelSet* é uma instrução que altera o valor predefinido da velocidade das instruções de movimento subsequentes. A utilização desta instrução necessita da definição de duas componentes da velocidade.

A primeira componente, corresponde a um valor percentual da velocidade predefinida, valor calculado com base na informação da posição com que são executados os próximos movimentos. A segunda componente, diz respeito à velocidade máxima do *TCP* em mm/s.

Para a aplicação desta instrução foram idealizadas duas abordagens. A primeira consiste em efectuar o procedimento de leitura da porta série (uso da rotina *GetData*) e, consequentemente, utilizar o *VelSet* com um valor de velocidade calculado em função da posição, antes de cada instrução de movimento.

Na segunda abordagem idealizada, com o intuito de aligeirar o código de programação, a rotina *GetData* é executada paralelamente ao programa principal a cada 0.1 segundos, à qual a instrução *VelSet* vai buscar o novo valor de velocidade actualizada, antes de cada instrução de movimento.

- ***SpeedRefresh***

O *SpeedRefresh* é uma instrução da linguagem *RAPID* que permite alterar a velocidade do movimento do robô durante a execução de um programa. Com esta instrução é possível criar uma adaptação "grosseira" da velocidade a partir de uma entrada externa, por exemplo um sinal de um sensor.

Tipicamente esta instrução é utilizada dentro de uma rotina que é executada paralelamente ao programa principal e calcula o valor de velocidade a ser actualizada. A velocidade é depois actualizada durante a execução do movimento.

Inicialmente deu-se prioridade à utilização da instrução *VelSet*, em detrimento do *SpeedRefresh*. Isto deve-se ao facto de estar referido no manual de instruções da *ABB,* que com esta última instrução, a actualização da velocidade não é instantânea podendo ter um atraso que pode variar entre os 0.3 e os 0.5 segundos, desde que é enviada a ordem de actualização da velocidade, até à sua influência física no manipulador.

Contudo, após um ensaio preliminar, verificou-se que a resposta do sistema com a instrução *SpeedRefresh*, era significativamente mais célere do que com a instrução *VelSet*, apesar do possível atraso de execução, anteriormente mencionado.

5.3 Ensaios de acabamento efectuados

Todo este processo de implementação culminou com a realização de um conjunto de ensaios, cada um deles focado num aspecto específico que se pretendeu testar e validar. De seguida descrevem-se detalhadamente todos os ensaios realizados, assim como os resultados obtidos após a sua execução.

- **Ensaios de actualização da velocidade (sem corte)**

Este grupo de ensaios inicia todo um conjunto de testes de validação e integração funcional do sistema desenvolvido. Para realizar estes ensaios, recorreu-se a uma peça cilíndrica de *MDF* (medium density fiberboard) com 100mm de raio que possui, em determinadas partes, sobre espessuras que o dispositivo teria que acomodar (Figura 5.12).

A escolha desta forma cilíndrica em disco, em detrimento de um sapato, prendeu-se com o facto de ser uma geometria cuja trajectória é de fácil parametrização. Uma vez que durante a fase do movimento, o robô apenas tem que executar um movimento rotacional do seu sexto eixo, este formato permite calcular, de uma maneira expedita, a velocidade efectiva do movimento do robô.

O diâmetro de 200mm do disco foi escolhido por conferir ao disco um perímetro de cerca de 630mm, aproximadamente idêntico ao de um sapato genérico número 40.

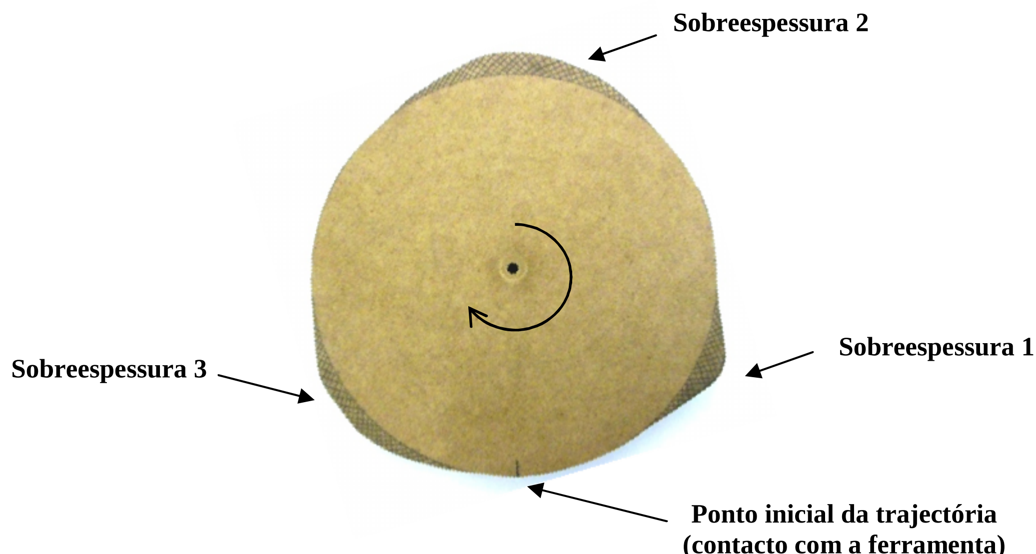
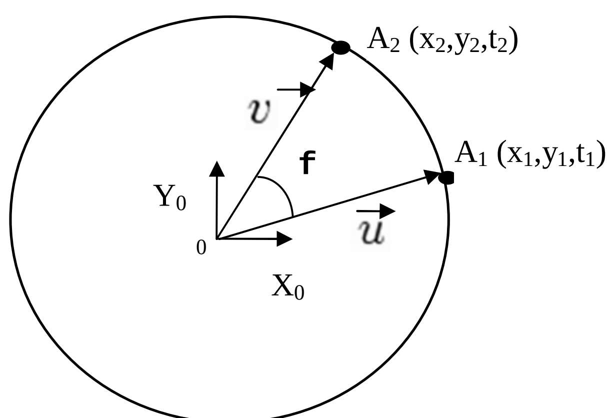


Figura 5.12 – Peça com sobreespessuras e sentido do movimento da trajectória

Assim, recorrendo a algumas ferramentas de programação do *RobotStudio*, é possível definir um qualquer ponto associado ao elemento terminal do robô, e registar a sua posição ao longo do tempo, em relação a um referencial fixo no espaço. Após o tratamento dos dados recolhidos torna-se possível calcular a velocidade real do movimento do robô (Figura 5.13), bem como o tempo de actualização da velocidade, desde que é dada uma determinada ordem.



$$\text{Velocidade} = \text{arco } A_1A_2 \, / \, (t_2 - t_1)$$

$$\text{arco } A_1A_2 = \mathbf{f} * r$$

$$\cos(\varphi) = \frac{u.v}{|u||v|}$$

Figura 5.13 – Cálculo da velocidade da trajectória

É importante realçar que, para efeitos de validação da arquitectura do sistema de acabamento desenvolvido, a escolha desta peça para a realização dos ensaios experimentais, em nada prejudica ou influencia os princípios assumidos e as soluções adoptadas para o acabamento das solas dos sapatos, que aqui se pretendem comprovar.

Actualização linear da velocidade

Nos primeiros ensaios deste tipo, utilizou-se uma actualização de velocidade linear, em que o novo valor de velocidade é calculado em função de uma equação cuja variável independente é a posição.

O novo valor de velocidade calculado é inversamente proporcional ao valor da posição medido pelo *encoder*. Considerando sobreespessuras até 1mm (valores de posição inferiores a 35 impulsos), o valor da velocidade é calculado com recurso à seguinte equação:

Velocidade = posição x (- (100 - 10) / (35 - 10)) + 136

O valor de velocidade obtido por esta expressão corresponde a uma percentagem da velocidade máxima predefinida.

O gráfico da Figura 5.14 evidencia o valor da velocidade em função da posição.

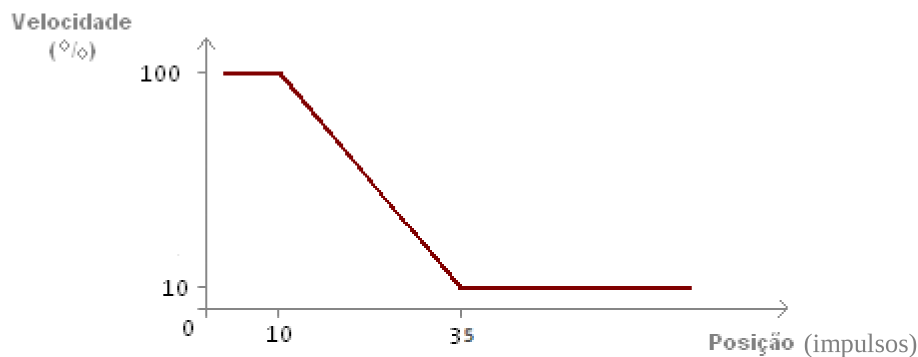


Figura 5.14 – Actualização linear da velocidade entre posição inicial e a posição 35

Para eventuais valores inferiores à posição inicial de referência cujo valor é 10, ou para valores de posição superiores a 35, foram impostas condições que mantém a velocidade actualizada a 100%, ou 10% respectivamente.

Realizou-se então um ensaio com uma velocidade predefinida de 100mm/s e uma força nominal de acomodação do dispositivo de 9N, e registaram-se os valores de posição a cada 0.1 segundos. De referir que a velocidade máxima predefinida de 100mm/s e a mínima de 10mm/s correspondem respectivamente aos 100% e 10% da velocidade actualizada, sendo directa a conversão de percentagem para mm/s. Os dados recolhidos e tratados apresentam-se na Figura 5.15.

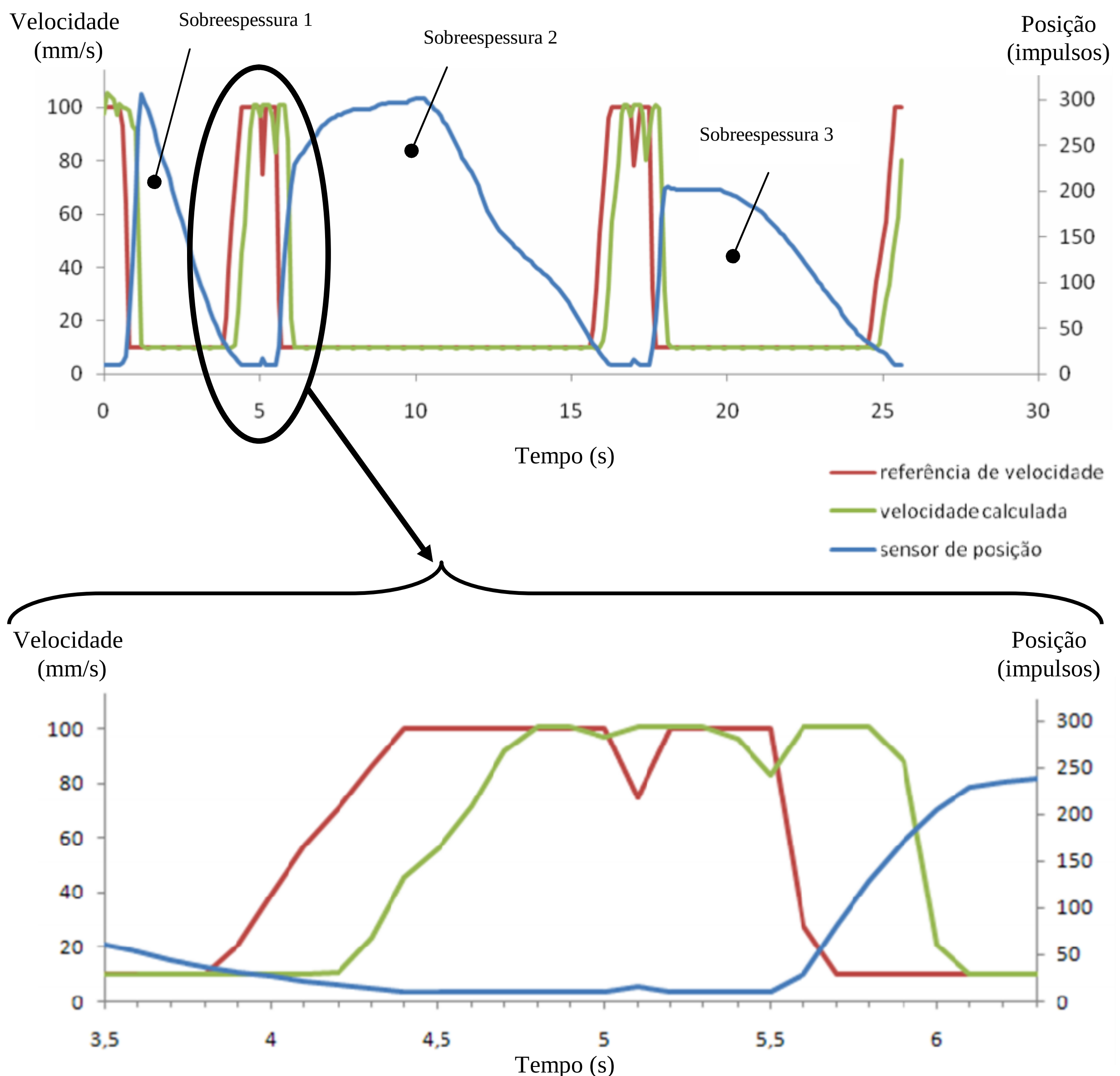


Figura 5.15 – Ensaio de actualização linear da velocidade entre a posição inicial e 35

Como é evidente na imagem, a curva a verde, correspondente à velocidade real, tem um desfasamento em relação à curva da referência de velocidade; desfasamento esse que corresponde sensivelmente a valores na ordem das 4 décimas de segundo. Ou seja, a actualização real da velocidade só ocorre 0.4 segundos depois da sua ordem de actualização. Pode ver-se claramente na imagem as sobreespessuras indicadas pelas curvas descritas pelos dados do sensor de posição. De notar que este atraso está de acordo com os dados fornecidos no manual de instruções, no que toca ao uso da instrução *SpeedRefresh*.

Com este método de actualização de velocidade o tempo de processamento do disco foi de 25.6 segundos.

Uma vez que neste ensaio a actualização de velocidade só ocorre linearmente para sobreespessuras até 1mm, e as sobreespessuras desta peça são bastante superiores, a rapidez com que é mensurado e ultrapassado este valor de posição é bastante elevada, sendo quase instantânea. Decidiu-se então aumentar a gama de valores para os quais a velocidade é actualizada linearmente.

Assim, como os valores máximos registados pelo sistema de medição rondam o valor 300, decidiu-se fazer uma actualização linear de velocidade dos 100% para os 10%, desde o valor inicial de referência, até à posição 300 (aproximadamente uma sobreespessura de 12mm) segundo a equação:

Velocidade = posição x (- (100 - 10) / (300 - 10)) + (2990/29)

Para todos os valores de posição superiores a 300, ou inferiores a 10, a velocidade mantém-se respectivamente nos 10% ou 100% da velocidade predefinida (Figura 5.16).

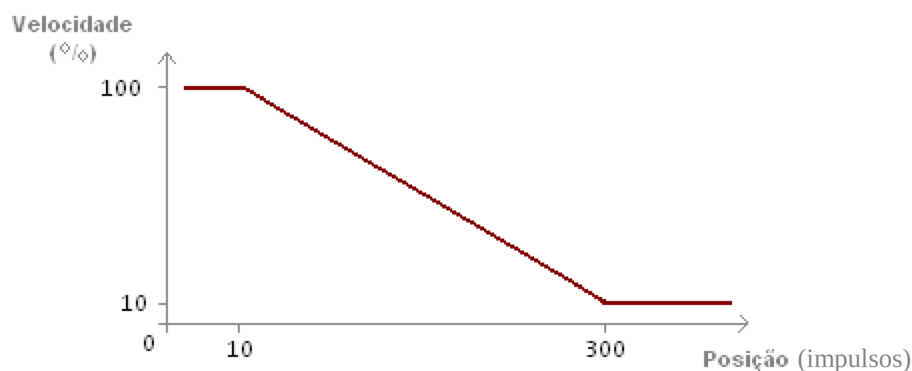


Figura 5.16 – Actualização linear da velocidade entre posição inicial e a posição 300

As restantes condições de realização do ensaio mantiveram-se as mesmas. Os resultados apresentam-se nos gráficos da Figura 5.17.

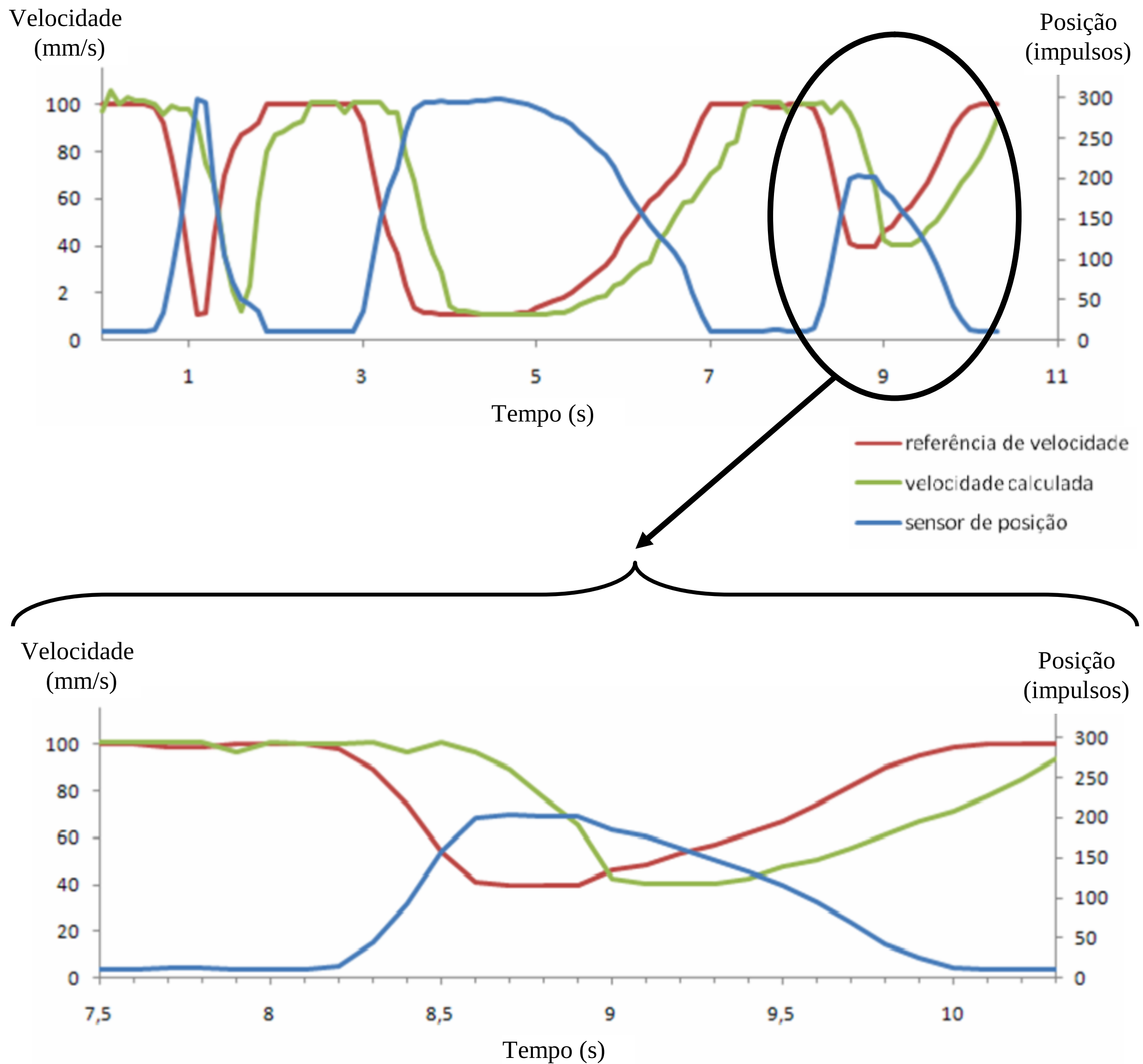


Figura 5.17 – Ensaio de actualização linear da velocidade entre a posição inicial e 300

Como seria de esperar, em termos de dinâmica de resposta do sistema, esta manteve-se inalterável, sendo o único ganho em termos do tempo de processamento do disco. O tempo necessário para efectuar a trajectória de corte foi reduzido para 10.3 segundos. Como é visível o atraso da actualização efectiva da velocidade manteve-se na casa dos 0.4 segundos.

Actualização de velocidade com 2 patamares

Após uma consulta aos manuais da *ABB*, referentes à aplicação do controlo de força pelo método *FCSpeedChange*, verificou-se que a actualização da velocidade neste modo de funcionamento, é realizada com base na definição de patamares de velocidade, para diferentes gamas de força, sendo a resposta temporal tanto menor quanto menor for o número desses patamares de velocidade e a razão entre eles.

Pensou-se então que, analogamente ao *FCSpeedChange*, uma alteração da velocidade com base na definição de patamares de velocidades para gamas de deslocamentos, pudesse ser uma opção viável em termos de ganhos na resposta temporal.

Como a Figura 5.18 mostra, foram então definidos dois patamares de velocidade, 100% e 10% da velocidade predefinida, e uma histerese determinada por um intervalo de valores de posição que delimitam dois estados.

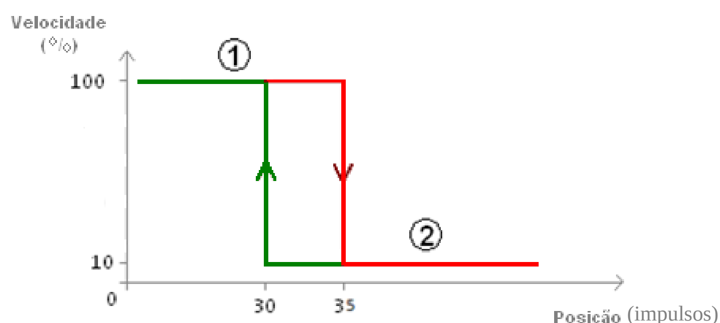


Figura 5.18 – Actualização da velocidade por patamares

Basicamente o sistema apresenta dois estados de velocidade, definidos com base no valor da posição, e numa variável booleana *flagpos*. Inicialmente a variável *flagpos* é inicializada por defeito como sendo falsa, e a velocidade é de 100% --» estado 1. Quando o *encoder* detecta valores superiores ou iguais à posição 35, a velocidade comuta para o valor de 10%, e a variável booleana torna-se verdadeira --» estado 2.

O sistema regressa repetidamente ao estado 1, sempre que a *flagpos* é verdadeira, e a posição medida pelo *encoder* é inferior a 30, readquirindo o sistema uma velocidade de 100%, e levando novamente a *flagpos* a tomar o valor de falso.

A histerese foi definida entre os valores de posição 30 e 35, para evitar a comutação intermitente de velocidade, em torno de um único valor de posição.

Com o intuito de melhorar a dinâmica do sistema, realizou-se o mesmo tipo de ensaio utilizando dois patamares, respectivamente 100% e 10% da velocidade predefinida. A força de acomodação do dispositivo manteve-se a mesma, 9N, e os valores do sensor, do tempo e da posição foram registados a cada 0.1segundos (Figura 5.19).

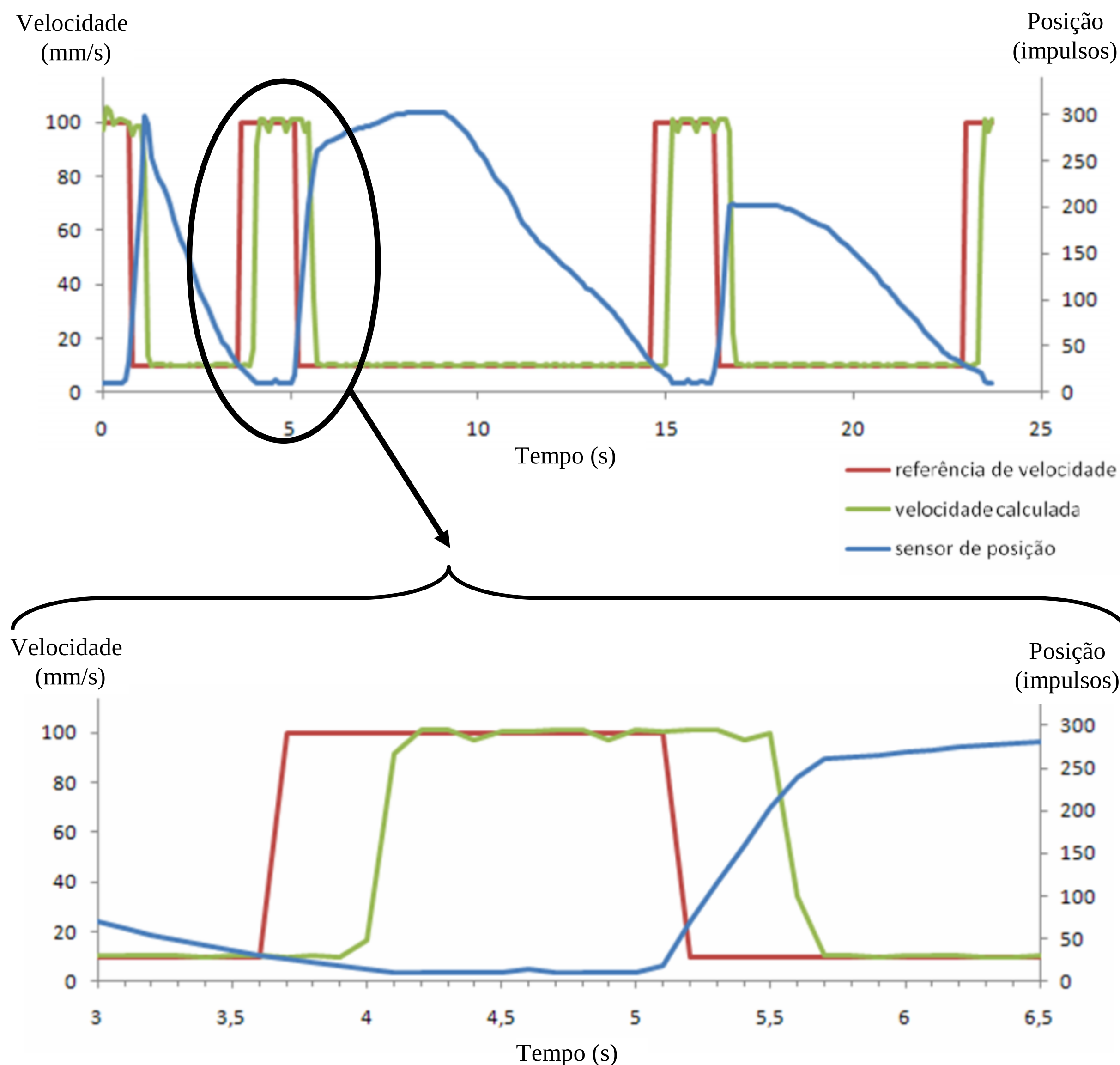


Figura 5.19 – Ensaio de actualização da velocidade por patamares

Realizados estes ensaios, não se verificou nenhuma melhoria na dinâmica da actualização da velocidade, mantendo-se o mesmo atraso verificado anteriormente, na ordem dos 0.4 segundos, o que seria de esperar.

Actualização de velocidade em pontos predefinidos

Com o intuito de se provar que o atraso detectado na actualização da velocidade por parte do robô não depende dos procedimentos da comunicação, e para melhor compreender a dinâmica do processo de actualização da velocidade, realizaram-se novos ensaios. Nestes testes, os novos valores de velocidade não foram ditados pelos erros geométricos da forma da peça, mas sim definidos ao nível da programação. Deste modo, os testes foram realizados sem que existisse contacto da peça com o dispositivo.

Sendo o contorno do disco definido com 24 pontos igualmente espaçados entre si, através dos quais é estabelecida uma trajectória que o robô cumpre ao executar um movimento de juntas do seu sexto eixo, foram atribuídos os valores de velocidade da seguinte maneira:

→ 3 pontos a 100% da velocidade de 100mm/s
→ 3 pontos a 10% da velocidade de 100mm/s
→ 3 pontos a 100% da velocidade de 100mm/s
→ 2 pontos a 10% da velocidade de 100mm/s
→ 2 pontos a 100% da velocidade de 100mm/s
→ 2 pontos a 10% da velocidade de 100mm/s
→ 1 ponto a 100% da velocidade de 100mm/s
→ 1 ponto a 10% da velocidade de 100mm/s
→ 1 ponto a 100% da velocidade de 100mm/s
→ 1 ponto a 10% da velocidade de 100mm/s
→ 1 ponto a 100% da velocidade de 100mm/s
→ 1 ponto a 10% da velocidade de 100mm/s
→ 1 ponto a 100% da velocidade de 100mm/s
→ 1 ponto a 10% da velocidade de 100mm/s
→ 1 ponto a 100% da velocidade de 100mm/s

A informação da velocidade da trajectória foi registada a cada 0.1 segundos.

O tratamento da informação recolhida neste ensaio apresenta-se sob a forma do gráfico da Figura 5.20.

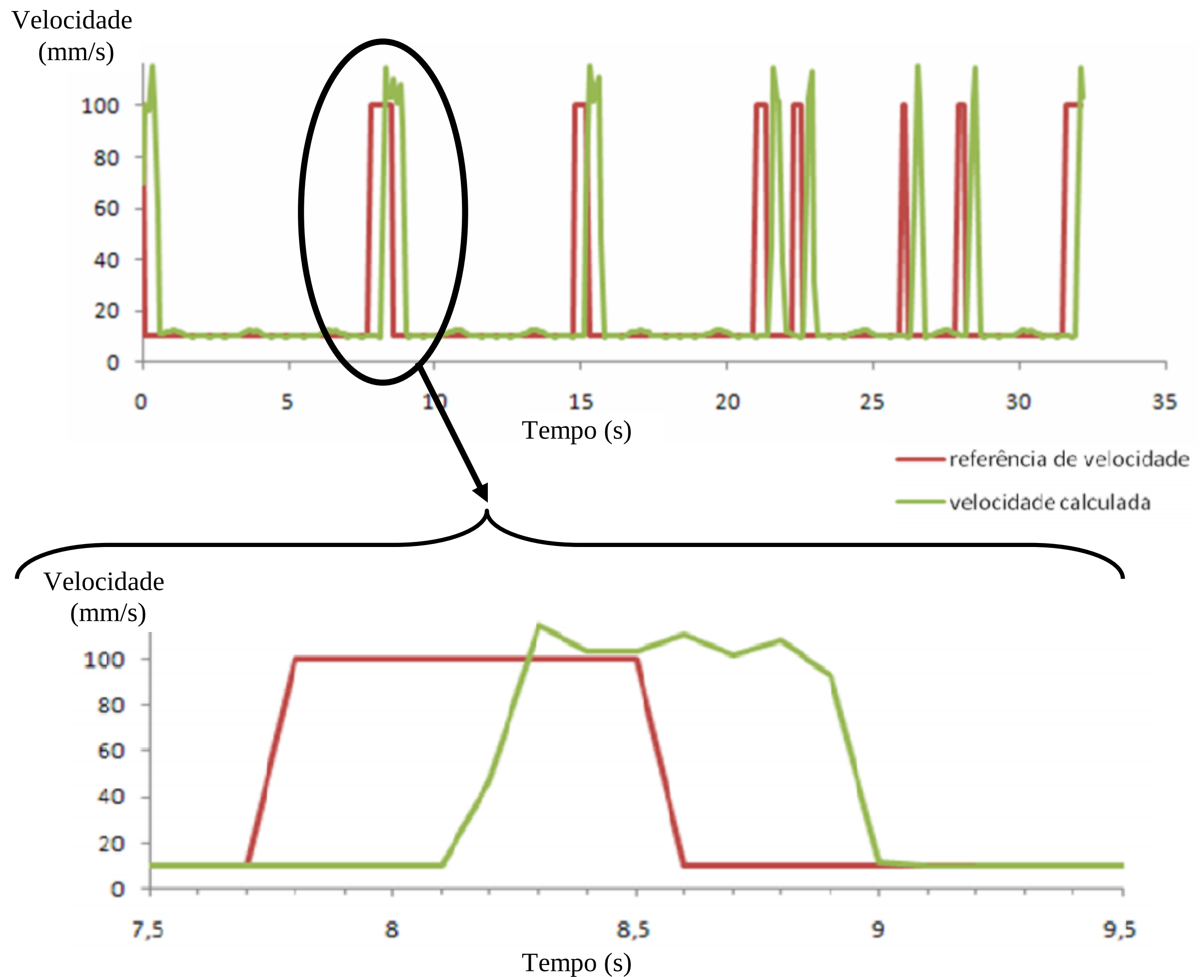


Figura 5.20 – Ensaio de actualização da velocidade em pontos pré definidos

Como se pode constatar, a curva a verde, correspondente à velocidade real, tem sempre um afastamento em relação à curva da referência de velocidade ordenada ao robô.

Analisados estes valores, chegaram-se a duas conclusões. A primeira tem a ver com o facto de a comunicação não interferir com o processo de actualização da velocidade uma vez que o tempo de actualização se manteve praticamente o mesmo, na casa dos 0.4 segundos. A segunda prende-se com o facto de para dinâmicas mais elevadas de comutação de velocidade, em que é ordenada uma transição impulsiva, o robô tenta cumprir o valor superior de velocidade, apesar de durante a sua aceleração, a sua referência já indicar um valor mais baixo. Este comportamento é evidenciado na Figura 5.20.

Pode ainda observar-se que para valores de 100% da velocidade máxima, que corresponde a 100 mm/s, é visível uma oscilação na curva da velocidade. Este comportamento pode dever-se ao facto de o robô, quando se movimenta a velocidades mais elevadas, iniciar o movimento a partir de um determinado ponto, com uma determinada aceleração, e reduzir a velocidade ao chegar ao ponto seguinte, para assegurar que cumpre a posição determinada pela definição dos parâmetros de zona na programação da trajectória.

- **Ensaios com a utilização do *FCSpeedChange***

Feitos os ensaios de actualização de velocidade do sistema desenvolvido, achou-se relevante realizar ensaios comparativos, utilizando desta vez, o sensor de força montado no robô utilizado no modo de controlo de força *FCSpeedChange*. Como já foi explicado anteriormente, neste modo de funcionamento, o robô realiza a trajectória predefinida, reduzindo a velocidade em função da força medida.

Assim, para se obter um método análogo de funcionamento ao da actualização de velocidade por patamares, foi definido o valor máximo e mínimo de velocidade, 100mm/s e 10mm/s respectivamente, e foi dada uma referência de força de 10N.

As pressões nas válvulas foram ajustadas para exercer uma força nominal idêntica à referência indicada de 10N, e os valores do sensor de posição, da força e da posição do disco foram registados a cada décima de segundo.

Apesar de não ter influência neste processo em causa, os valores do sensor de posição foram registados, para avaliar a coerência da medição da força em função das sobreespessuras.

Os dados recolhidos e tratados, apresentam-se sob a forma dos gráficos da Figura 5.21.

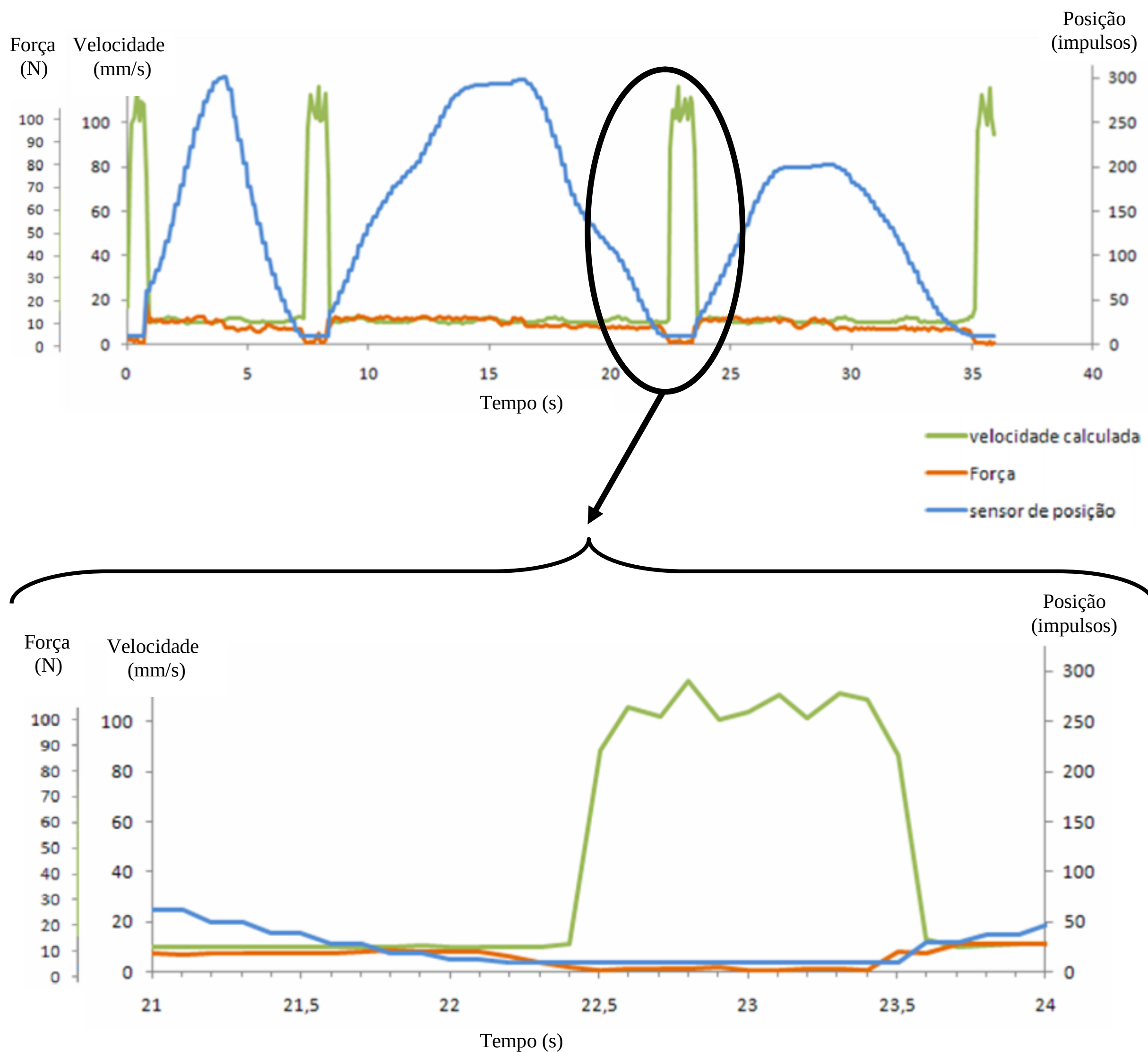


Figura 5.21 - Ensaio com a utilização do *FCSpeedChange*

Da análise deste gráfico podemos tirar várias conclusões. Quanto à actualização da velocidade, podemos ver que esta é praticamente instantânea, nos momentos em que a força se anula ou regista valores superiores à referência indicada, respectivamente, aumentando ou reduzindo a velocidade.

Podemos também ver que quando a curva do sensor de posição tem um comportamento horizontal, correspondente a deslocamento nulos, a força sentida é sensivelmente zero. Isto comprova a coerência entre os valores registados pelo sensor de posição do dispositivo, e os valores registados pelo sensor de força do robô. Ou seja, na ausência de deslocamento, a força medida tende para um valor nulo, sendo que o eventual

contacto tangencial entre a ferramenta e a peça não provoca deslocamentos nem, consequentemente, forças associadas significativas. Nesses intervalos de tempo, a trajectória descrita corresponde à trajectória exacta pré-definida.

É também visível na figura que a força se mantém praticamente constante quando os deslocamentos não são nulos, ou seja quando há contacto entre as sobreespessuras da peça e o dispositivo, evidenciando assim a capacidade de manutenção de uma força relativamente constante por parte do dispositivo.

- **Ensaios de corte com actualização da velocidade**

Tendo sido testada a dinâmica de funcionamento do sistema e obtidos os resultados anteriormente apresentados, realizaram-se ensaios para avaliar a capacidade de corte do dispositivo. Para isso utilizou-se uma placa de borracha com 3mm de espessura, apertada entre dois discos de *MDF*, com 100mm de raio, pretendendo simular a sola em elastómero de um sapato. Como a Figura 5.22 evidencia, a placa de borracha não é totalmente circular, contendo sobre espessuras em determinadas zonas.



Figura 5.22 – Peça utilizada para ensaios de corte

A força nominal de acomodação do dispositivo foi 9N, e a ferramenta utilizada foi uma vulgar fresa para madeira com 2 dentes rectos e 12mm de diâmetro.

Após alguns ensaios concluiu-se que uma passagem não era suficiente para remover a totalidade da sobreespessura, sendo necessárias duas a três passagens, para se obter um acabamento satisfatório.

Inferiu-se que, apesar do atraso referido e calculado anteriormente da actualização da velocidade, a incapacidade de remoção do material excedentário deveu-se sobretudo à

limitada capacidade de corte, causada pela reduzida potência da ferramenta de corte, e pela desadequação da fresa ao processo de maquinagem em questão.

Mais uma vez se enfatiza a importância fulcral que a correcta definição dos parâmetros de corte tem neste processo. Ou seja, tendo este sistema um atraso real na actualização do valor de velocidade, no intervalo de tempo que decorre entre a detecção de uma sobreespessura pelo sistema de medição, até que a velocidade tenha uma actualização efectiva, o mecanismo de corte terá que ter capacidade para remover uma determinada quantidade de material, o que não se verificou.

Apesar de em ambiente industrial não ser aceitável repetir o processo de corte por questões de ineficiência de produtividade, o aspecto final da peça ao fim de 3 passagens é o apresentado na Figura 5.23.

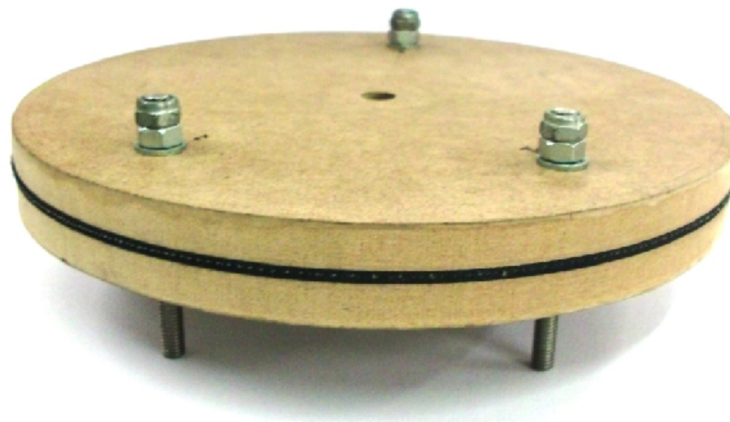


Figura 5.23 – Qualidade do acabamento da peça utilizada para testes de corte

Com a obtenção deste resultado, pode-se acreditar que com um processo de corte adequado, se obtenha um método mais robusto e veloz do ponto de vista da exequibilidade deste tipo de operações em ambiente industrial numa só operação.

5.4 **Conclusão**

Finda a implementação do sistema desenvolvido e realizados os ensaios entendidos como necessários para avaliar o seu comportamento, podemos retirar várias conclusões.

Quanto à capacidade do dispositivo exercer uma força supostamente constante, os resultados relevaram-se bastante satisfatórios, tendo em conta que o controlo de força é

realizado de uma maneira passiva, não contemplando factores importantes como a força de atrito do mecanismo, ou a inércia provocada pela massa dos componentes móveis do mesmo.

Quanto à actualização da velocidade do sistema desenvolvido, durante todos os ensaios realizados, existe sempre um atraso real de cerca de 0.4 segundos que não se conseguiu eliminar. Não foi possível determinar com maior resolução este atraso verificado, uma vez que os dados utilizados para os cálculos foram registados a cada 0.1 segundos. Como foi provado, esse atraso prende-se com limitações das instruções de programação do código *RAPID*, e não com toda a dinâmica de comunicação e cálculo de nova velocidade do sistema desenvolvido.

Nos ensaios realizados no modo *FCSpeedChange*, os atrasos na actualização de velocidade não se verificam, pelo que é possível crer que, as limitações temporais das instruções de programação utilizadas talvez sejam de carácter propositado.

Apesar de depender bastante do processo de maquinagem de cada peça, em termos temporais, a utilização de uma actualização de velocidade linear torna-se proveitosa, visto reduzir o tempo de processamento da peça.

Quanto à capacidade de corte do mecanismo, esta revelou-se insuficiente para cumprir o objectivo numa só passagem, mas tal deve-se ao facto de a ferramenta de corte não ser apropriada nem possuir a potência de accionamento adequada.

6 . Conclusões e Trabalhos Futuros

6.1 Conclusões

Este projecto assentou no desenvolvimento de uma solução robótica para operações de acabamento de sapatos, baseado na concepção de um dispositivo auxiliar para acabamento de baixo custo, com capacidade de regulação da força de contacto e sua integração numa célula robotizada.

Foi desenvolvido um dispositivo auxiliar de acabamento com capacidade de regulação da força de contacto, ao longo de um grau de liberdade linear segundo o qual é feita a acomodação das sobreespessuras das peças a maquinar. Possui um curso máximo ajustável de 52mm, consegue regular forças de contacto até 50N, contém um sistema electrónico de medição de posição da ferramenta com uma resolução de 0.042mm, e comunicação dos respectivos dados via porta série.

A regulação da força de contacto apesar de se realizar de uma maneira passiva, na qual as válvulas pneumáticas são ajustadas previamente para o valor nominal de força pretendido, apresentou um comportamento bastante satisfatório, para as dinâmicas de movimento do dispositivo envolvidas neste processo em particular. Para este facto, também contribuiu a solução mecânica adoptada e o uso de um actuador pneumático de baixo atrito.

Para além do projecto mecânico do dispositivo e do projecto electrónico de monitorização da posição da ferramenta com interface de comunicação série, este trabalho envolveu ainda o desenvolvimento de *software* de comunicação e a programação de um robô industrial.

Neste sistema desenvolvido, o robô previamente programado no modo *off-line*, é controlado em posição e consegue fazer uma adaptação de velocidade em função do sinal de posição enviado pelo dispositivo. Contudo, nessa actualização, está implícito um atraso de cerca de 0.4 segundos. Esse atraso é inerente às instruções da linguagem de programação do robô, e não devido à dinâmica do *software* de comunicação desenvolvido para interligar o controlador do robô com o dispositivo. Apesar de não ter sido possível reduzir ou eliminar este atraso, a adaptação da velocidade da trajectória do robô revelou-se satisfatória para o processo em causa.

A definição do modo e dos parâmetros de actualização da velocidade podem ser facilmente alterados, em função do acabamento pretendido, ou da temporalidade do processo, consoante os requisitos das peças em questão.

A capacidade de corte da ferramenta transportada pelo dispositivo revelou-se insuficiente, dada a sua inadequação, quer ao nível da ferramenta de corte, quer ao nível da potência de accionamento.

Desde que seja correctamente configurado, o dispositivo desenvolvido, dadas as suas características, pode ter uma vasta gama de aplicações que envolvam o controlo de força numa direcção, representando uma mais-valia económica relativamente a soluções comerciais, que possuem, em muitos casos, potencialidades supérfluas ou subaproveitadas na maior parte dos processos industriais.

6.2 **Sugestões de trabalhos futuros**

Aproveitando as potencialidades do dispositivo construído e identificadas as limitações do sistema desenvolvido, com o intuito de posteriormente o aperfeiçoar, várias alterações podem ser efectuadas.

Com algumas modificações no sistema pneumático, o controlo da força poderá ser feito de uma maneira activa, contemplando assim a inércia do dispositivo durante as acelerações e a força de atrito. Para isso seria necessário substituir as válvulas existentes, por válvulas proporcionais de elevado desempenho, com transdutores de pressão, e utilizando as potencialidades do sistema de medição, alterar a programação do micro controlador para calcular acelerações e actuar sobre as válvulas.

A capacidade de corte do mecanismo deve ser melhorada, adoptando-se uma ferramenta eléctrica com as características adequadas, podendo ser regulada a velocidade da ferramenta, em função do sinal de erro de posição, com recurso à programação do micro controlador, em cooperação com o *driver* da ferramenta eléctrica. Isto potencia a capacidade de corte do sistema, resolvendo de certa maneira o atraso de 0.4 segundos na actualização da velocidade efectiva da trajectória por parte do robô.

Poderá ser também elaborado um estudo de um mecanismo de preensão e orientação das peças no robô, uma vez que o seu impreciso posicionamento no robô condiciona

irremediavelmente a correcta execução da trajectória tida como exacta e, consequentemente, todo o processo de acabamento em questão.

Numa perspectiva mais ambiciosa poderá pensar-se em elaborar um estudo de um mecanismo de alimentação e posicionamento automático das peças na célula robotizada.

7 Bibliografia

ABB. *ABB Robotics.* www.abb.com/robotics. [Online].

ABB. *ABB Robotics.* www.abb.com/review [Online].

ABB. Force Control for Machining. *Application Manual.* Revision B, 2007.

ABB. Machining PowerPac. *Operating Manual*, 2008.

ABB. RAPID. *Reference Manua*, 2007.

Abreu, Paulo. *Aplicaçações industriais de robôs.* Faculdade de Engenharia da Universidade do Porto. Porto. Textos de Apoio, 2001.

Abreu, Paulo. *Apontamentos de robótica.* Faculdade de Engenharia da Universidade do Porto. Porto. Textos de Apoio, 2008.

APICCAPS – *Monografia estatística '07*. 2007. Publicação Online, 2009.

Arvy Plast. www.arvyplast.com.br. [Online] 2009.

Autec Automation. www.autecautomation.com [Online] 2009.

Desma. www.desma.de [Online] 2009.

Distretto Calzaturiero Veneto. *Trasferimento tecnológico per l'Automazione nel Settore Calzaturiero*. Progetto di Ricerca: Relazione Finale, 2008.

Ferraresi, D. *Usinagem dos metais*. Editora Edgard Blucher, cop. 1995.

Hu, Zhongxu, Marshall, Chris, Bicker, Robert e Taylor, Paul. *Automatic surface roughing with 3D machine vision and cooperative robot control* School of Mechanical and Systems Engineering, University of Newcastle, Publicação Online (www.sciencedirect.com), 2007.

Jatta, Francesco, Zanoni, Lorenzo, Fassi, Irene e **Negri, S.** *A roughing/cementing robotic cell for custom made shoe manufacture***,** International Journal of Computer Integrated Manufacturing, Publicação online, 2008.

Kochan, Anna. Actis and the shoe industry. *Assembly Automation.* 1996, Vol. 16, 3, pp. 30-31.

Magalhães, António Pessoa. *Comunicações entre computadores*. Faculdade de Engenharia da Universidade do Porto. Textos de Apoio, 2001.

MikroElektronika. www.mikroe.com [Online] 2009.

Microchip. www.microchip.com [Online] 2009.

Moita, Nuno. *Concepção e Simulação de uma Célula Robotizada para Acabamentos de Solas de Calçado,* Dissertação de Mestrado Integrado em Engenharia Mecânica na opção de Automação, Faculdade de Engenharia da Universidade do Porto. Porto, 2009.

Morais, J. M. S. *Desenho técnico básico 3*. 23ª ed, Porto Editora.

Nemec, Bojan e Zlajpah, Leon. *Automation in Shoe Assembly,* Jozef Stefan Institute. Ljubljana, Slovenia.

PushCorp. www.pushcorp.com [Online] 2009.

Robot System. www.robotsystem.it [Online] 2009.

Rooks, Brian W. Robots bring automation to shoe production, *Assembly Automation* 1996. Vol. 16, 3, pp. 22–25.

Santos, João. *Desenvolvimento de um sistema de acabamento de solas para a indústria do calçado,* Dissertação de Mestrado Integrado em Engenharia Mecânica na opção de Automação, Faculdade de Engenharia da Universidade do Porto. Porto, 2009.

São Paulo Design. *Diagnóstico sectorial – sector do calçado,* www.spdesign.sp.gov.br. Publicação Online.

Sepúlveda, João, Couto, Carlos e Monteiro, João. *Developing an Automated System for Shoe Sole Halogenation,* Department of Industrial Electronics, University of Minho.

Shih, Alber J., Lewis,Mark A. e Strenkowski, John S. *End Milling of Elastomers — Fixture Design and Tool Effectiveness for Material Removal*. Journal of Manufacturing Science and Engineering. Fevereiro, 2004 Vol 116, pp. 115-123.

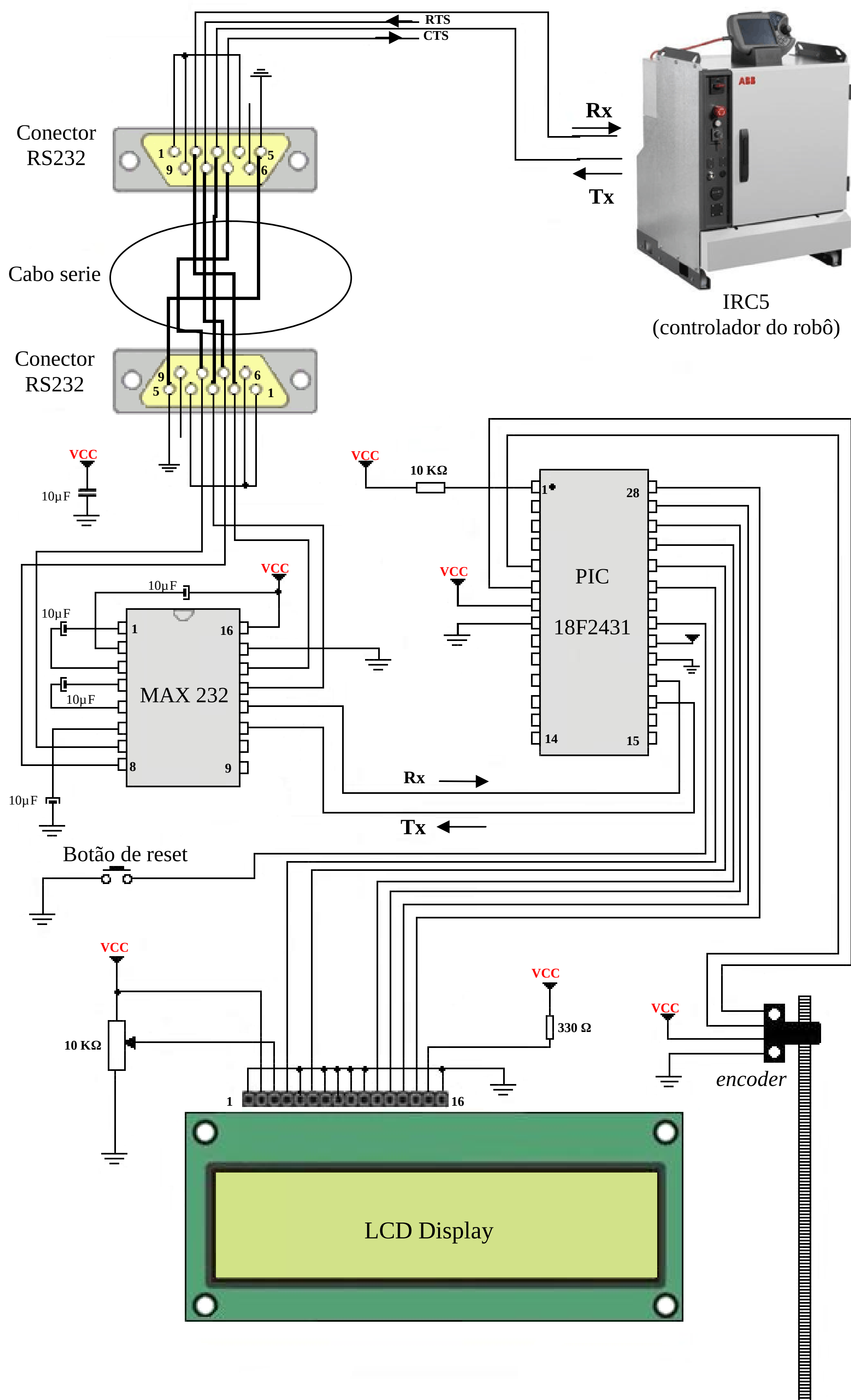
SMC. *European Best Pneumatics,* Catálogo electrónico v3.1.

SMC www.smc.eu. [Online] 2009.

Spencer Jr, James E. Robotics technology and the advent of agile manufacturing systems in the footwear industry. *Assembly Automation* 1996. Vol. 16, 3, pp. 5-15.

Anexos

Anexo A – Diagrama de ligações do sistema electrónico



Anexo B – Código de programação do micro controlador PIC 18F2431

```
//----------------------------------------------------------------------------------------------------------//
#include "built_in.h"
//-------------------------------//Definição das rotinas//--------------------------------------------------//
void escreve_apagando_LCD(char linha, char coluna,char *text);
void config();
//--------------------------------//Definição das variáveis globais//-------------------------------------//
void main ()
{
char oldstate=0;
char txt1[8];
char txt2[8];
unsigned int posi1;
unsigned int posi2;
unsigned short bytemais;
unsigned short bytemenos;
unsigned short bmais;
unsigned short bmenos;
unsigned short loint;
unsigned short bmaismais;
	config();
	OSCCON=0b01111110;
	ANSEL0=0;
	QEICON=0b10110100;
	DFLTCON=0b00111000;
	PIR3.F3=0;
```

```
LCD_Cmd(LCD_CLEAR);
Usart_Init(19600);
        while (1)    // A//
           {
                bytemais=POSCNTH;
                bytemais&=0b00111111;
                posi1=bytemais*256+POSCNTL;
                bytemenos=POSCNTL;
                bytemais<<=1;
                bytemais&=0b01111110;
                loint=bytemenos&0b10000000;
                loint=loint>>7;
                loint&=0b00000001;
                bytemais=(bytemais|0b10000000)|loint;
                bytemenos=bytemenos&0b01111111;
                Usart_Write(bytemais);
                Usart_Write(bytemenos);
                intToStr(posi1,txt1);
                LCD_OUT(1,1,txt1);
             if (button(&portb,0,1,0)) oldstate=1;
              if (oldstate && button(&portb,0,1,1))
                {
                 POSCNTH=0;
                 POSCNTL=0b00001010;
                 oldstate=0;
                }
```

```
            }   //Fim do ciclo while (A)//

      }   //Fim da rotina main //

//-----------------------------------------// Rotina de configuração//----------------------------------//

          void config()

{         TRISB  = 0;

          LCD_Init(&PORTB);

    OSCCON=0b01111110;

          LCD_Cmd(LCD_CURSOR_OFF);

          TRISA  = 255;

          TRISC=0b00111111;

          INTCON2.f7=0;

          TRISB  = 0b00000001;

          escreve_apagando_LCD(1,2,"Projecto Final");

          PORTC  = 0;

          Delay_ms (200);

          LCD_OUT(2,3,"Diogo Viana");

          Delay_ms (2000);

}  //Fim da rotina config (1)//

//---------------------------------------//Rotina escrve_apagando_LCD //-----------------------------//

void escreve_apagando_LCD(char linha,char coluna,char *text)

{         LCD_Cmd(LCD_CLEAR);

         LCD_OUT(linha,coluna,text);

 }
```

Anexo C – Codificação e reconstrução da informação da posição

Codificação da informação da posição no micro controlador

A codificação desta informação que foi realizada ao nível do micro controlador, aquando da sua programação é descrita de seguida. Torna-se no entanto conveniente abordar sumariamente os operadores binários utilizados.

Operador “e”

A operação binária “e”, aqui representada por ”&”, compara dois pares de *bits*, um bit de cada *byte*, e retorna o valor 0 (zero) sempre que haja um *bit* que tome o valor 0 (zero). Caso contrário o resultado da operação é 1 (um).

&	0	1
0	0	0
1	0	1

Operador “ou”

A operação binária “ou”, aqui representada por ”|”, compara dois pares de *bits*, um bit de cada *byte*, e retorna o valor 1 (um) sempre que haja um *bit* que tome o valor 1 (um). Caso contrário o resultado da operação é 0 (zero).

|	0	1
0	0	1
1	1	1

Operador “left shift”

Esta operação binária, aqui representada por ”<<”, provoca um translação para a esquerda n casas dos *bits*, e atribui o valor de 0(zero) aos bits menos significativos. Exemplifica-se de seguida a operação.

byte1 = 00101111

byte1 = byte1<<1

byte1= 01011110

Operador "shift right"

Esta operação binária, aqui representada por ">>", provoca um translação para a direita de n casas dos *bits*, e atribui o valor de 0(zero) aos bits mais significativos . Exemplifica-se de seguida a operação.

byte1 = 00101111
byte1 = byte1>>2
byte1= 00001011

- **Codificação dos *bytes high* e *low***

As operações de codificação enumeram-se de seguida. As letras de A a P, utilizadas na representação simulam *bits* genéricos, que podem tomar o valor 0 (zero) ou 1 (um).

- *byte high* = ABCDEFGH

Este b*yte* é um valor medido directamente pelo *encoder* e lido pelo micro controlador, através da instrução "*POSCNTH*".

- *byte high* &= 00111111

O resultado desta operação será um *byte* com a composição genérica **00CDEFGH**. A composição deste *byte* pode ser entendida como a composição fundamental do *byte high*, uma vez que os *bits* descartados são supérfluos, dada a resolução do sistema de medição e o curso de medição, e uma vez que toda a informação efectiva da posição estará contida em 14 *bits*, e não em 16, sendo estes dois *bits* restantes utilizados para a codificação dos *bytes.*

- *byte low* = **IJKLMNOP**

Este byte é também um valor medido directamente pelo *encoder* e lido pelo micro controlador, através da instrução "POSCNTL".

- posição = *byte high* x 256 + *byte low*

A posição medida pelo *encoder* é calculada segundo a fórmula acima descrita.

- byte high <<= 1

Esta instrução faz uma translação de uma casa para a esquerda, dos *bits*, atribuído o valor de 0(zero) ao *bit* menos significativo. O resultado desta operação é *byte high* = 0CDEFGH0.

- lointr = *byte low* & 10000000

Este byte denominado “lointr”, é um byte intermédio utilizado para concluir a codificação do *byte high*, através do *byte low*. O resultado desta operação “e” será lointr = I0000000.

- lointr = lointr >> 7

Agora é realizada uma translação para a direita de 7 casas, dos *bits*, sendo o resultado do *byte* lointr = 0000000I.

- byte high = (byte high | 1000 0000) | loint

O operador “ou” é representado por ”|” e retorna o valor 1, sempre que haja um 1 em qualquer posição dos *bits*. O resultado da primeira operação “ou” dentro de parêntesis será 1CDEFGH0, e este valor tomara a forma de 1CDEFGHI com a realização do segundo “ou”, ficado o *byte high* finalmente codificado.

- byte low = byte low & 01111111

A codificação do *byte low* fica concluída com esta operação sendo o seu resultado 0JKLMNOP

Terminada a codificação dos *bytes high* e *low*, temos a informação da posição do sensor contida em dois *bytes*, apresentando uma forma do tipo:

1CDEFGHI 0JKLMNOP

Como desejado o *bit* mais significativo do *byte high* toma o valor 1 (um), e o *bit* mais significativo do *byte low* toma o valor 0 (zero). A informação efectiva da posição medida pelo *encoder* está contida nos *bits* representados aqui pelas letras de C a P (14 *bits*). Ou seja, com esta codificação conseguimos obter valores de posição que podem variar de 0 (zero) a 16383 (2^{14} – 1(valor zero)). Dada a resolução da medição, esta amplitude de escala é mais do que suficiente para o curso de trabalho do mecanismo que se construiu.

É preciso ter em atenção que estes *bytes* terão que ser reconstruídos ao nível do controlador do robô, após a sua identificação, realizando as operações inversas às aqui demonstradas.

Leitura da porta serie e reconstrução da informação da posição no controlador do robô

- **Leitura da porta série**

Para efectuar a leitura dos *byte*s enviados via porta séries, primeiramente é necessário definir ao nível do controlador do robô, todas as variáveis utilizadas no procedimento de leitura e tratamento dos dados, *GetData*. Assim, o canal de comunicação é definido como uma variável ou dispositivo de *Input/output*, os *bytes* lidos tem que ser definidos como *bytes*, os valores como variáveis numéricas, etc. Depois, foram acrescentadas instruções para abrir o canal de comunicação em modo binário, limpar o *buffer* associado, e ler dois *bytes* antes de se fechar o canal de comunicação.

Lidos estes *bytes* torna-se necessária a sua identificação e descodificação para se poder reconstruir correctamente o valor da posição. Recorde-se que estes *bytes* foram codificados previamente aquando da programação do micro controlador.

Após a leitura, podem ocorrer três situações:

→ O *byte high* é o primeiro *byte* lido, e o *low* é o segundo;

→ O *byte low* é o primeiro *byte* lido e o *high* é o segundo;

→ Ambos os *bytes* são *bytes high*, ou *bytes low*.

Para isso, e sabendo que o bit mais significativo do *byte high* tem o valor “1”, e o *bit* mais significativo do *byte low* tem o valor “0”, é necessário testar cada um dos *bytes* para determinar qual dos dois é o *high*, e qual é o *low*. O teste consiste em fazer uma operação binária de comparação de bits do tipo “e”, com o *byte* 10000000, que corresponde ao valor numérico na escala decimal de 128 (2^7=128).

O resultado da operação binária efectuada, será 128 para o *byte high* e 0 (zero) para o *byte low*.

byte high	1	C	D	E	F	G	H	I	
128	1	0	0	0	0	0	0	0	
Resultado da operação “e”	1	0	0	0	0	0	0	0	(128)

byte low	0	J	K	L	M	N	O	P	
128	1	0	0	0	0	0	0	0	
Resultado da operação "e"	0	0	0	0	0	0	0	0	(0)

A identificação dos *bytes* é feita com recurso a uma variável booleana definida por defeito como "falsa", que apenas toma o valor de "verdadeira" quando o primeiro *byte* lido é o *high* e o segundo é o *low*, ou vice-versa, mantendo-se como falsa caso ambos sejam *high* ou ambos sejam *low*, prevendo assim as três situações hipotéticas acima descritas

Após a identificação dos *bytes*, e caso a variável booleana adquira o valor de verdadeira, é realizada então a sua descodificação para a posterior reconstrução da posição.

- **reconstrução do *byte low***

Para fazer a reconstrução do *byte* low, recorre-se à utilização de um *byte* intermédio, que se forma a partir do resultado da operação "e" do *byte high* com o *byte* 00000001, e ao qual seguidamente se executa uma operação *shift left* de 7 casas, seguindo-se um "ou" com o *byte low*.

byte high	1	C	D	E	F	G	H	I
1	0	0	0	0	0	0	0	1
Resultado da operação "e"	0	0	0	0	0	0	0	I

byte intermédio	0	0	0	0	0	0	0	I
Resultado da operação "<<7"	I	0	0	0	0	0	0	0

byte intermédio	I	0	0	0	0	0	0	0
byte low	0	J	K	L	M	N	O	P
Resultado da operação "ou"	**I**	**J**	**K**	**L**	**M**	**N**	**O**	**P**

O *byte low* fica então reconstruído e tem a sua forma primordial **IJKLMNOP.**

- **reconstrução do *byte high***

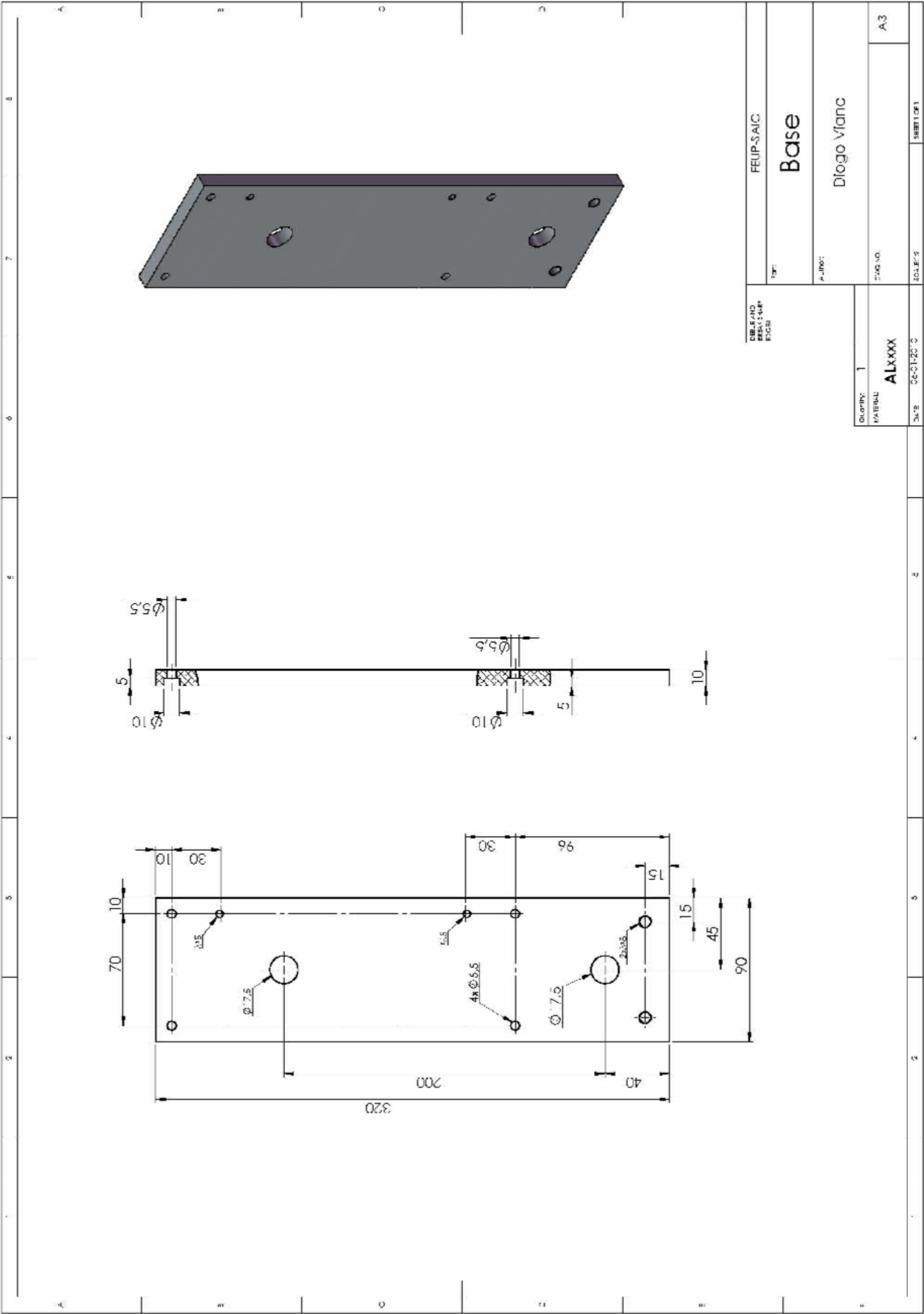
Para reconstruir o *byte high,* executam-se sequencialmente operações do tipo “e”com o *byte* 01111110, seguindo-se a operação *right shift* de 1 casa.

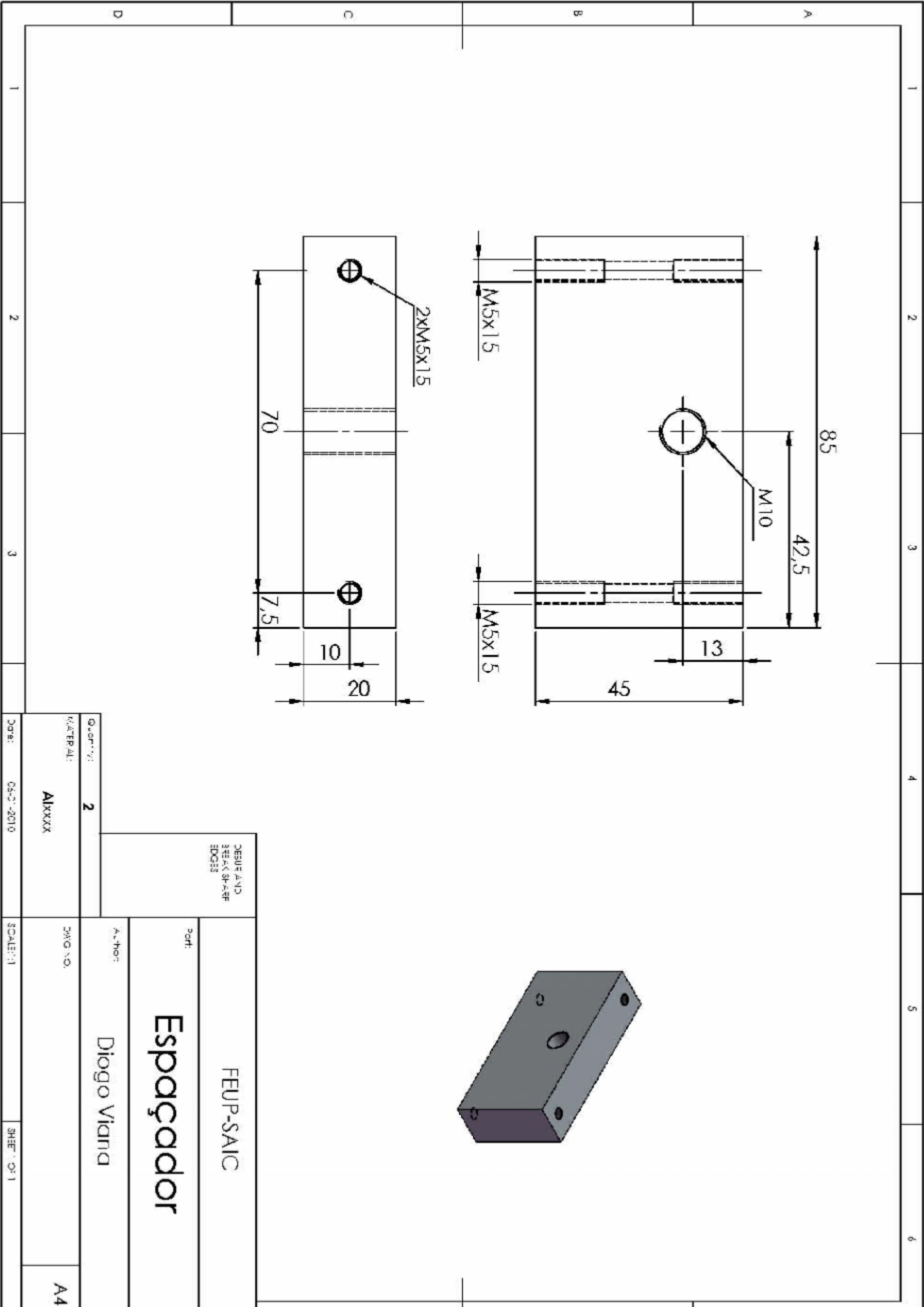
byte high	1	C	D	E	F	G	H	I
126	0	1	1	1	1	1	1	0
Resultado da operação “e”	0	C	D	E	F	G	H	0

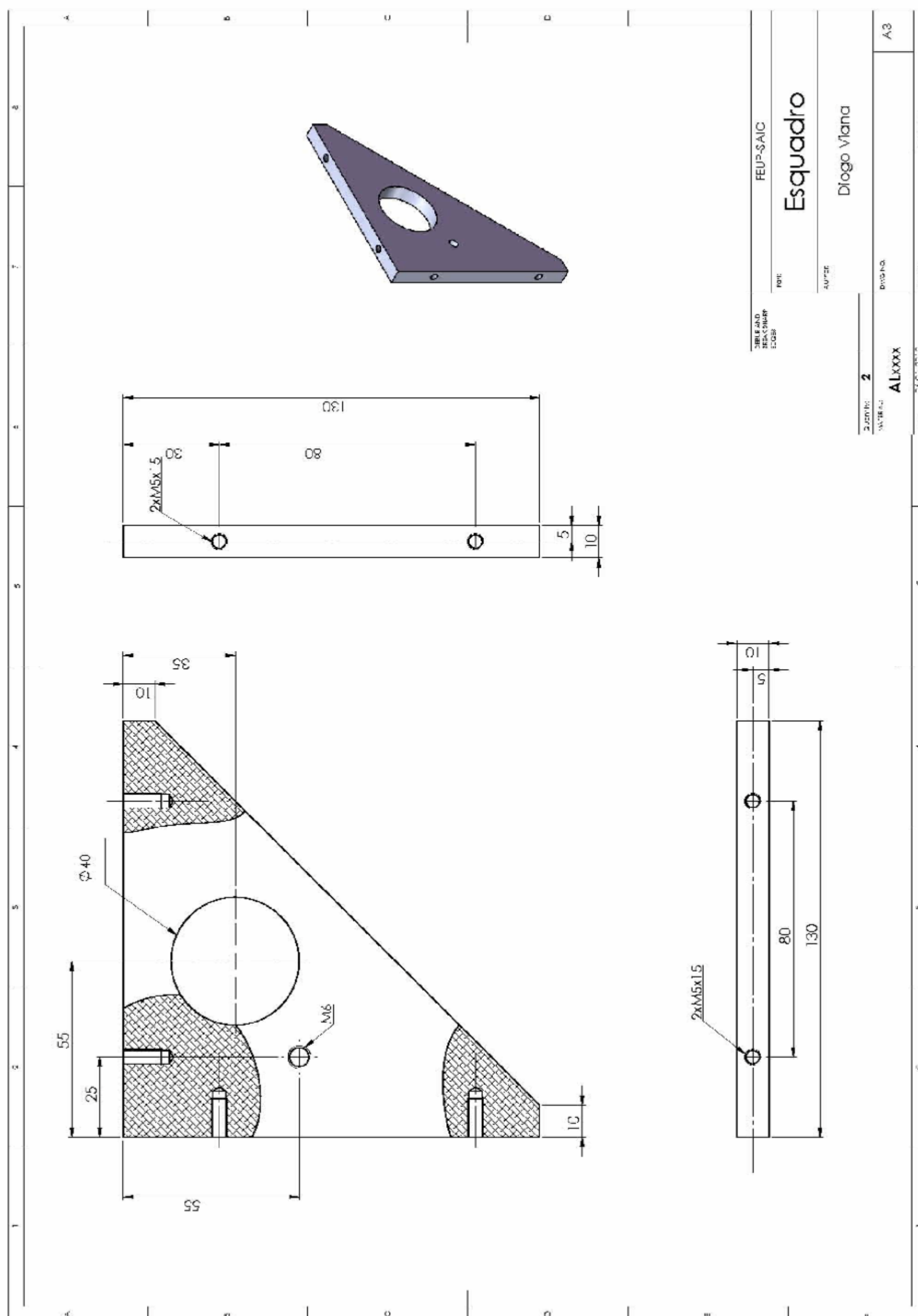
byte high	0	C	D	E	F	G	H	0
Resultado da operação “>>1”	0	0	C	D	E	F	G	H

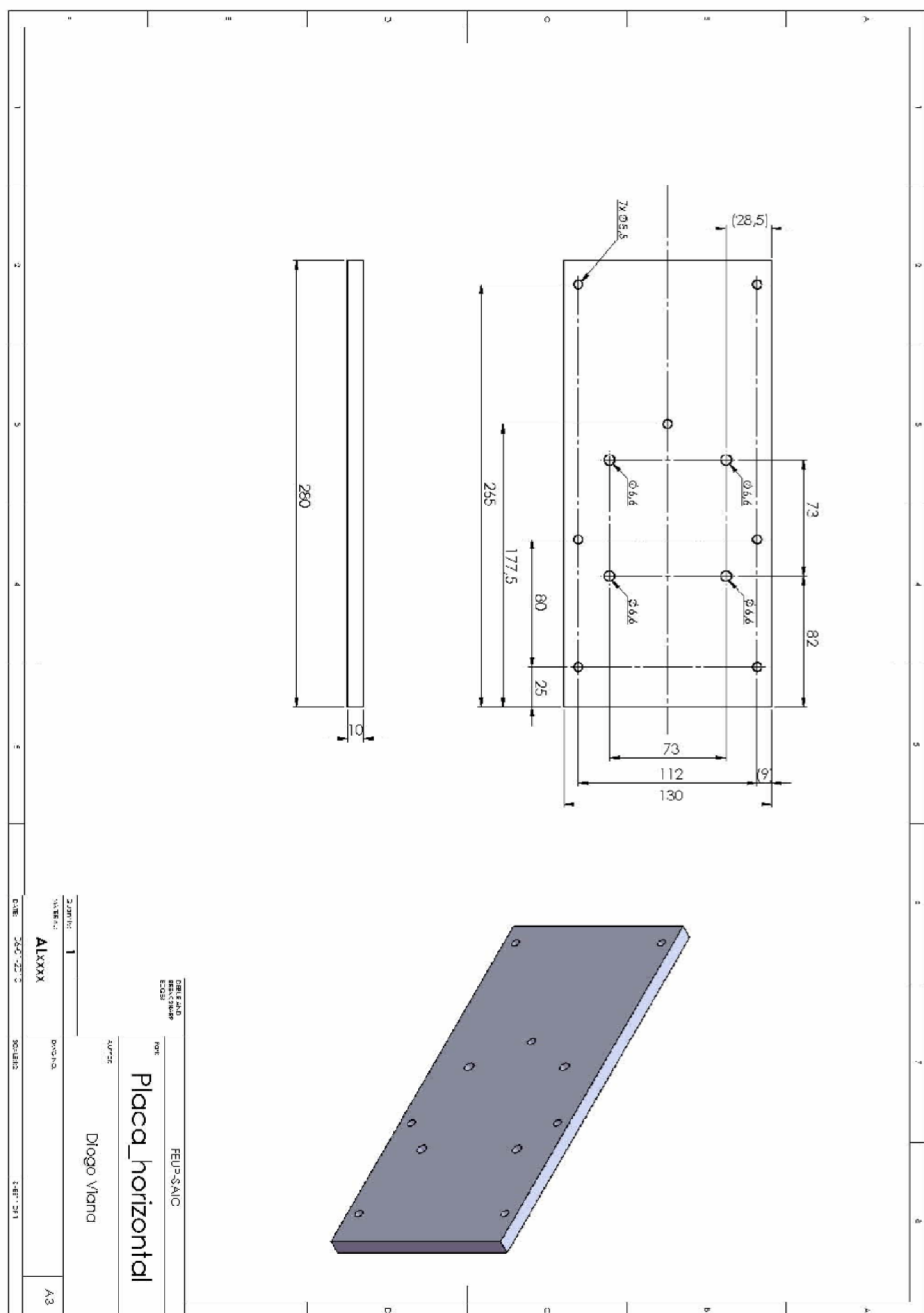
Feita a reconstrução dos *bytes*, o valor da posição pode então ser calculado e utilizado convenientemente na actualização da velocidade da trajectória.

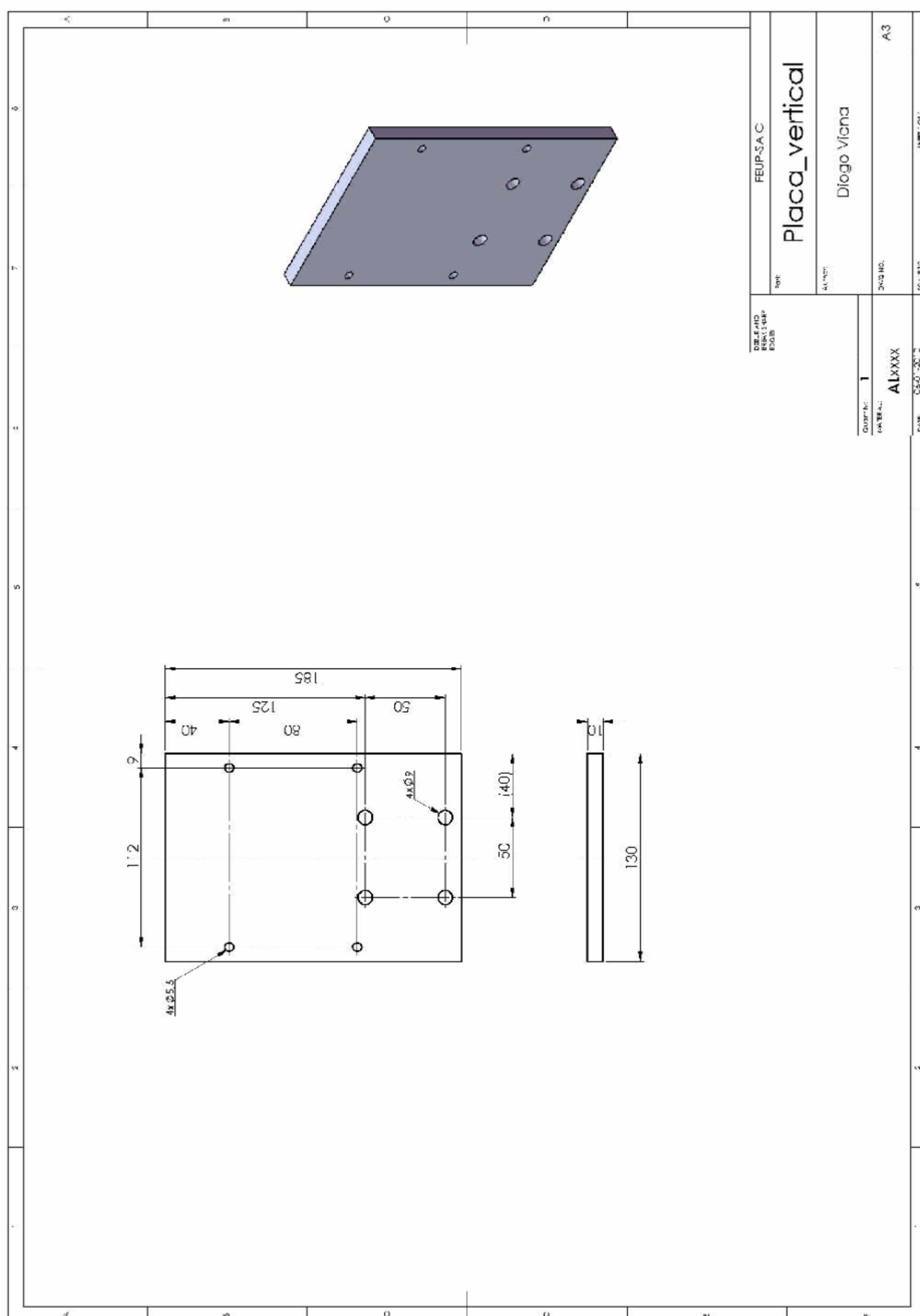
Anexo D – Desenhos de definição das peças maquinadas

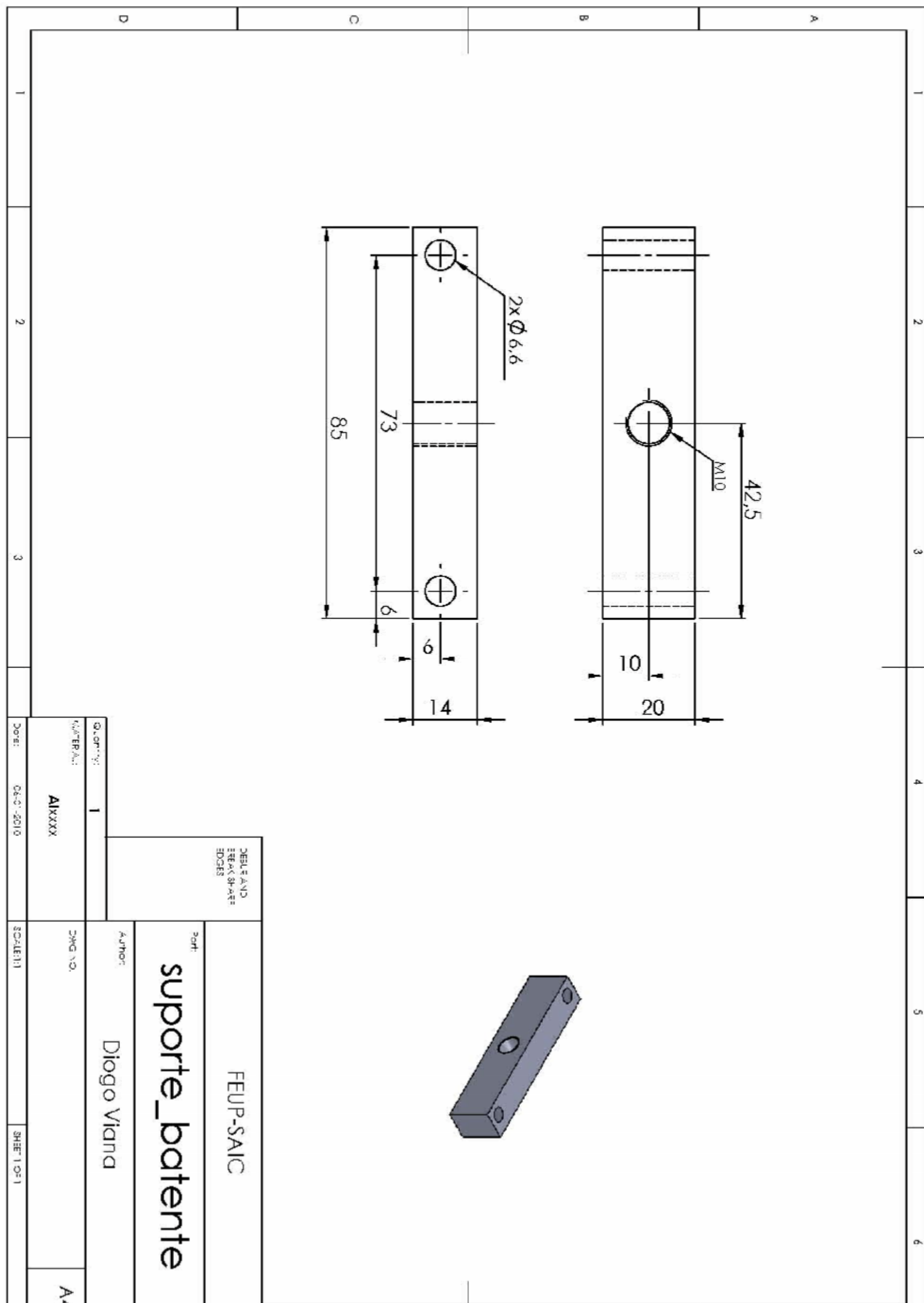



2xM5x15
70
7,5
10
20
M5x15
M5x15
M10
85
42,5
13
45
2
Alxxxx
FEUP-SAIC
Espaçador
Diogo Viana
A4


FEUP-SAIC
Esquadro
Diogo Viana
A3
2
ALxxxx
130
30
80
2xM5x15
5
10
35
10
Ø40
M6
55
25
10
55
10
5
80
130
2xM5x15


7x Ø5,5
(28,5)
Ø6,6
Ø6,6
Ø6,6
Ø6,6
280
265
177,5
80
25
73
82
10
73
112
130
(9)
FEUP-SAIC
Placa_horizontal
Diogo Viana
ALxxxx
1
A3


FEUP-SAC
Placa_vertical
Diogo Viana
A3
ALXXXX
185
125
50
40
80
4xØ9
4xØ5.5
50
130
10


2xØ6.6
85
73
6
6
14
M10
42,5
10
20
Alxxxx
1
FEUP-SAIC
suporte_batente
Diogo Viana
A4

Anexo E – Código de programação *RAPID* do robô

```
MODULE MainModule

    TASK PERS tooldata Tooldata_1:=[FALSE,[[935.4,-561,533],[0.887011,0,0,0.461749]],[1,[0,0,1],[1,0,0,0],0,0,0]];
    PERS wobjdata Wobj_S_1:=[TRUE,TRUE,"",[[0,0,0],[1,0,0,0]],[[4.44089E-12,-2.95823E-28,340],[1,0,1.11022E-16,0]]];
    TASK PERS tooldata Tooldata_3:=[TRUE,[[21.362881370527,-102.803829206644,360.000000000003],[7.55022998913941E-09,0.971567081760922,-0.236764451805091,2.4784905733415E-09]],[1,[0,0,1],[1,0,0,0],0,0,0]];
    TASK PERS wobjdata Workobject_3:=[FALSE,TRUE,"",[[849.389078715741,-500.77445496402,582.400030507226],[1,0,0,0]],[[0,0,0],[1,0,0,0]]];
    PERS wobjdata Wobj_s_2:=[TRUE,TRUE,"",[[0,0,0],[1,0,0,0]],[[4.44089209850063E-13,1.80411241501588E-13,140.000000000005],[-1.94289029309402E-16,0,1,0]]];
    CONST robtarget p26:=[[154.999999999996,-40,20.0000000000032],[7.85046229341887E-17,0.707106781186548,-0.707106781186547,7.85046229341887E-17],[-1,0,-2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget p24:=[[104.999999999996,-2.57167334302694E-14,20.0000000000031],[7.85046229341887E-17,-0.707106781186547,0.707106781186548,7.85046229341887E-17],[-1,0,-2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget p22:=[[89.7140374773768,-54.5563147537205,20.0000000000032],[1.03487107482808E-16,-0.490110217178018,0.871660470032751,-1.10046107405013E-17],[-1,0,-2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget p21:=[[71.6680800379543,-76.7377762492031,20.0000000000028],[2.23768290674688E-16,-0.366854218813057,0.930278443337833,-2.09254141135008E-18],[-1,0,-2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget p2:=[[101.106315171515,28.3286609714876,20.000000000003],[-8.28811669218476E-18,0.796805111415905,-0.604236389521095,9.57923485296758E-17],[-1,0,1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget p7:=[[-7.16545340329477,104.755220765007,20.000000000003],[-4.18959953171079E-17,0.999417022366174,0.0341411101859682,4.16576489010305E-17],[-1,0,1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget p16:=[[-60.5514338220654,-85.7818387660963,20.0000000000032],[1.74720000037773E-16,0.302514550881076,0.953144766814161,-3.27918874189777E-17],[-1,0,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget p20:=[[48.306828961767,-93.2279479322495,20.0000000000032],[8.57035275849918E-17,-0.236764420466447,0.971567089397941,2.28318305239841E-17],[-1,0,-2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
```

```
	CONST	robtarget
p5:=[[48.3068289617672,93.2279479322495,20.0000000000031],[-
1.20372723812651E-17,0.971567089397942,-
0.236764420466447,8.57035275849919E-17],[-
1,0,1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
	CONST	robtarget	p27:=[[134.999999999996,-
20,20.000000000003],[7.85046229341887E-17,0.707106781186548,-
0.707106781186547,7.85046229341887E-17],[-
1,0,2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
	CONST	robtarget	p8:=[[-
35.1623592779575,98.9373968224762,20.0000000000031],[-4.22943382287217E-
17,0.985459517717197,0.169910385028666,8.44953296912418E-17],[-
1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
	CONST	robtarget	p9:=[[-
60.5514338220652,85.7818387660968,20.0000000000031],[-6.0618855572151E-
17,0.953144766814161,0.302514550881076,1.89280000040921E-16],[-
1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
	CONST	robtarget	p10:=[[-
81.4496855239684,66.2642341542355,20.0000000000029],[-1.18671125557408E-
16,0.903074732324533,0.429483443030081,1.22938111862284E-16],[-
1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
	CONST	robtarget	p11:=[[-
96.3071866580771,41.8321144338554,20.0000000000029],[-5.60121417949766E-
17,0.836182124254711,0.548451871249319,9.12813497550076E-17],[-
1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
	CONST	robtarget	p12:=[[-
104.022024333819,14.2974981551059,20.0000000000032],[6.97462196541058E-
18,0.753713025327361,0.657203678817972,3.22219834971368E-17],[-
1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
	CONST	robtarget	p13:=[[-104.022024333819,-
14.2974981551059,20.000000000003],[5.52376859950917E-
17,0.657203678817972,0.753713025327361,-1.52763395340381E-16],[-1,0,-
1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
	CONST	robtarget	p14:=[[-96.3071866580769,-
41.8321144338554,20.0000000000028],[7.46847407086426E-
17,0.548451871249319,0.836182124254711,-3.65818364183438E-17],[-1,0,-
1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
	CONST	robtarget	p28:=[[114.999999999996,-4.93038065763132E-
29,20.0000000000035],[7.85046229341887E-17,0.707106781186548,-
0.707106781186547,7.85046229341887E-17],[-
1,0,2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
	CONST	robtarget
p3:=[[89.7140374773768,54.5563147537206,20.000000000003],[-
1.99783361280937E-17,0.871660470032751,-
0.490110217178018,1.43289841130041E-16],[-
1,0,1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
	CONST	robtarget	p17:=[[-35.1623592779577,-
98.9373968224763,20.0000000000033],[1.12660439588322E-
16,0.169910385028666,0.985459517717197,-7.45230835971311E-17],[-1,0,-
1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
	CONST	robtarget	p18:=[[-7.165453403295,-
104.755220765007,20.000000000003],[5.55435318680406E-
17,0.034141110185968,0.999417022366174,-7.31391120398558E-17],[-1,0,-
1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
```

```
CONST robtarget p19:=[[21.3628813705222,-
102.803829206644,20.0000000000031],[1.11607429090603E-16,-
0.102264148942035,0.994757278858095,2.59767204610504E-17],[-1,0,-
2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget
p6:=[[21.3628813705222,102.803829206644,20.0000000000031],[-
1.76209630285157E-17,0.994757278858095,-
0.102264148942035,9.76565004542773E-17],[-
1,0,1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget p1:=[[104.999999999996,7.39557098644699E-
29,20.0000000000032],[7.85046229341887E-17,0.707106781186548,-
0.707106781186547,7.85046229341887E-17],[-
1,0,2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget
p4:=[[71.6680800379537,76.7377762492032,20.0000000000028],[1.19866869023661
E-18,0.930278443337833,-0.366854218813057,1.9393251858473E-16],[-
1,0,1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget p23:=[[101.106315171515,-
28.3286609714876,20.0000000000032],[6.09587672461574E-17,-
0.604236389521095,0.796805111415904,1.08656684990352E-16],[-1,0,-
2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget p15:=[[-81.4496855239686,-
66.2642341542355,20.0000000000031],[1.69039903810641E-
16,0.429483443030081,0.903074732324533,-5.75722793209049E-17],[-1,0,-
1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget
p25:=[[107.999999999996,2.99999999999997,20.0000000000032],[7.8504622934188
7E-17,-0.707106781186547,0.707106781186548,7.85046229341887E-17],[-1,0,-
2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget
p29:=[[117.999999999996,2.99999999999997,20.0000000000033],[7.8504622934188
7E-17,-0.707106781186547,0.707106781186548,-7.85046229341888E-17],[-1,0,-
2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget
p30:=[[137.999999999996,23,20.0000000000033],[7.85046229341887E-17,-
0.707106781186547,0.707106781186548,-7.85046229341888E-17],[-1,0,-
2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget
p31:=[[157.999999999996,43,20.0000000000037],[7.85046229341887E-17,-
0.707106781186547,0.707106781186548,-7.85046229341888E-17],[-1,0,-
2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];


CONST robtarget p23_2:=[[101.106315171515,-
28.3286609714877,20.0000000000032],[6.0958833685941E-17,-
0.604236389521095,0.796805111415904,1.0865667066292E-16],[-
1,0,2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget p22_2:=[[89.7140374773768,-
54.5563147537206,20.0000000000031],[1.03487107482808E-16,-
0.490110217178018,0.871660470032751,-1.10046392068731E-17],[-
1,0,2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget p26_3:=[[154.999999999996,-40,20.0000000000032],[-
7.85046463303895E-17,0.707106781186548,-
0.707106781186547,7.85045913493176E-17],[-
1,0,2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
```

```
	CONST	robtarget	p27_2:=[[134.999999999996,-20,20.0000000000031],[-7.85046463303895E-17,0.707106781186548,-0.707106781186547,7.85045913493176E-17],[-1,0,2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
	CONST	robtarget	p136:=[[-197.182872471104,33.2350033139943,-209.999999999996],[0.691749956329962,1.2257458303267E-16,-1.32909794045459E-16,-0.722137104653607],[-1,0,-2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
	CONST	robtarget	p137:=[[-178.060619963469,12.3941909729593,-209.999999999996],[0.691749956329962,1.2257458303267E-16,-1.32909794045459E-16,-0.722137104653607],[-1,0,-2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
	CONST	robtarget	p138:=[[-158.938367455833,-8.4466213680756,-209.999999999996],[0.691749956329962,1.2257458303267E-16,-1.32909794045459E-16,-0.722137104653607],[-1,0,-2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
	CONST	robtarget	p97:=[[-148.947601243666,-8.87626132642549,-209.999999999996],[0.691749956329962,1.21321874844354E-16,-1.14101804322045E-16,-0.722137104653607],[-1,0,-2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
	CONST	robtarget	p98:=[[-147.83831113675,-19.153613165348,-209.999999999996],[0.794409261846106,9.97640758174833E-17,-1.52856530192243E-16,-0.607382848533876],[-1,0,-2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
	CONST	robtarget	p99:=[[-143.633938391757,-28.5295892902126,-209.999999999996],[0.87704298710817,4.59846580092707E-17,-1.81968913879683E-16,-0.480411905310827],[-1,0,-2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
	CONST	robtarget	p100:=[[-136.992685357181,-36.3693434455757,-209.999999999996],[0.933182290837863,3.24485036070552E-17,-1.93329045431847E-16,-0.359403411317419],[-1,0,-2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
	CONST	robtarget	p101:=[[-128.57138640758,-42.2005813030151,-209.999999999996],[0.971875059692382,1.45552436145377E-17,-2.14190924071151E-16,-0.235497066537841],[-1,0,-2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
	CONST	robtarget	p102:=[[-108.608371558029,-47.9897227033579,-209.999999999996],[0.997415379385476,-1.69280027395966E-17,-1.66964994861396E-16,-0.0718509635657499],[-1,0,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
	CONST	robtarget	p103:=[[-87.4096898402135,-50.1389013474575,-209.999999999996],[0.998994358980619,-2.22072611018136E-17,-1.66701095153035E-16,-0.0448360426989592],[-1,0,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
	CONST	robtarget	p104:=[[-66.2110593559146,-51.0128521818411,-209.999999999996],[0.999575754690897,-3.61136109997487E-17,-2.08255168395505E-16,0.0291257726785563],[-1,0,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
	CONST	robtarget	p105:=[[-45.3282902859458,-46.8298692864122,-209.999999999996],[0.988416349292605,-6.04049368462877E-17,-2.10606408186387E-16,0.15176666449217],[-1,0,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
	CONST	robtarget	p106:=[[-24.7521262570094,-40.2045792858178,-209.999999999996],[0.991007059933827,-6.04054477309167E-17,-
```

```
1.6804466933353E-16,0.133809592934556],[-1,0,-
1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget p107:=[[-3.10094562545871,-37.3943556340015,-
209.999999999995],[0.999890022207505,-2.74469287023673E-17,-
2.22069027586467E-16,-0.0148304918959001],[-1,0,-
1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget p108:=[[18.5444239184354,-40.2107135838158,-
209.999999999996],[0.995422089325069,-1.16116241339467E-17,-
2.23065780141151E-16,-0.0955764829009446],[-1,0,-
2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget p109:=[[39.731696903349,-44.0447827141446,-
209.999999999996],[0.997741440547836,-1.69424322471758E-17,-
1.94728835962474E-16,-0.0671715550923914],[-1,0,-
2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget p110:=[[61.0497223719813,-45.4822503501859,-
209.999999999996],[0.999990305214951,-2.96187416350309E-17,-
1.66535068215463E-16,-0.00440334828381367],[-1,0,-
1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget p111:=[[82.5208395629817,-44.8542915593208,-
209.999999999995],[0.999595712474382,-3.61141119698363E-17,-
1.3883400693528E-16,0.0284325799538698],[-1,0,-
1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget p112:=[[103.840611925257,-42.6426506286676,-
209.999999999996],[0.995834979085942,-4.9955620686876E-17,-
2.22973293355132E-16,0.0911739789024351],[-1,0,-
1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget p113:=[[123.813384159776,-35.6637630002736,-
209.999999999996],[0.964715962258461,-7.84016119006848E-17,-
1.58238975883204E-16,0.263292825887321],[-1,0,-
1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget p114:=[[132.118826506089,-29.2221394351928,-
209.999999999996],[0.923127212663715,-1.24430261846416E-16,-
1.65367962066101E-16,0.384494667374882],[-1,0,-
1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget p115:=[[138.491735979319,-20.8584826858597,-
209.999999999995],[0.865415281865971,-1.21058600438872E-16,-
1.44323878821536E-16,0.501055276304764],[-1,0,-
1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget p116:=[[142.369475096595,-11.1306819281891,-
209.999999999996],[0.783919522050102,-1.46097459984895E-16,-
1.68179233283351E-16,0.620862450909007],[-1,0,-
1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget p117:=[[143.17722807752,-0.657933984201597,-
209.999999999996],[0.681480610837426,-1.60042008176357E-16,-
1.71822694242974E-16,0.731836168177447],[-
1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget p118:=[[143.17722807752,-0.657933984201729,-
209.999999999996],[0.681480610837434,-1.61314768874451E-16,-
1.52731283771531E-16,0.73183616817744],[-
1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget p119:=[[141.088850585392,9.32516266506711,-
209.999999999996],[0.576671697945317,-1.83632736025614E-16,-
1.68457226184027E-16,0.816975980545857],[-
1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
```

```
CONST robtarget p120:=[[136.384803500714,18.3283336499168,-209.999999999996],[0.452792907513582,-1.94600438789716E-16,-1.22597218971661E-16,0.891615714815187],[-1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget p121:=[[129.113434972723,25.3539113246505,-209.999999999996],[0.291447478099918,-2.42100736144383E-16,-9.52335418943421E-17,0.95658683218472],[-1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget p122:=[[119.935194845853,29.7967723318495,-209.999999999996],[0.164984896652067,-2.066736029412E-16,0,0.986296093410446],[-1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget p123:=[[99.6145288348895,33.7328702006001,-209.999999999995],[0.0549162162547845,-2.28353771667223E-16,0,0.998490966004329],[-1,0,1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget p124:=[[78.765535550855,35.7510081109391,-209.999999999996],[0.0456169436506766,-1.98221445172535E-16,0,0.998959005391098],[-1,0,1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget p125:=[[-4.53560577830059,41.4430382888868,-209.999999999996],[0.0306823261752803,-1.8049229791737E-16,-4.52305595362433E-16,0.999529186597607],[-1,0,1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget p126:=[[-46.1459158168152,43.3404204525887,-209.999999999996],[0.0119196046617071,1.55971848694377E-16,-1.16428255816217E-15,-0.999928958988942],[-1,0,1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget p127:=[[-66.7873372339451,41.4171738694048,-209.999999999996],[0.0772532560071578,1.50872853754056E-16,0,-0.997011501656973],[-1,0,1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget p128:=[[-87.4160994805062,37.1719333058785,-209.999999999996],[0.122004485985679,2.07043294670133E-16,1.13748176517406E-16,-0.992529548879715],[-1,0,1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget p129:=[[-107.912187898806,31.43782660078,-209.999999999996],[0.146340380582617,1.71246521504461E-16,0,-0.989234296317579],[-1,0,1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget p130:=[[-127.588190869192,24.0576856722858,-209.999999999996],[0.237498348960989,1.74048758339886E-16,0,-0.971387942194469],[-1,0,1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget p131:=[[-136.163711170992,18.3659651152592,-209.999999999996],[0.347502728504979,2.13122340204734E-16,-3.99357664543219E-17,-0.937678971547083],[-1,0,1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget p132:=[[-142.809149967018,10.6990364913952,-209.999999999996],[0.480877732513997,1.61406860800508E-16,-7.21482139298814E-17,-0.876787663218522],[-1,0,1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget p133:=[[-147.141177071284,1.33411252967238,-209.999999999996],[0.591884859591925,1.48422395099442E-16,-9.37870775568241E-17,-0.806022526351371],[-1,0,1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget p134:=[[-148.947601243666,-8.87626132642424,-209.999999999996],[0.691749956329887,1.21321874844372E-16,-1.14101804322058E-16,-0.72213710465368],[-1,0,2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
```

```
CONST robtarget p135:=[[-152.073723094821,-11.7445992025689,-209.999999999996],[0.691749956329887,1.21321874844372E-16,-1.14101804322058E-16,-0.72213710465368],[-1,0,2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget p139:=[[-162.064489306989,-11.3149592442169,-209.999999999996],[0.691749956329887,1.22574583032688E-16,-1.32909794045474E-16,-0.72213710465368],[-1,0,2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget p140:=[[-182.905301648028,-30.4372117518479,-209.999999999996],[0.691749956329887,1.22574583032688E-16,-1.32909794045474E-16,-0.72213710465368],[-1,0,2,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget p141:=[[-203.746113989067,-49.5594642594788,-209.999999999996],[0.691749956329887,1.22574583032688E-16,-1.32909794045474E-16,-0.72213710465368],[-1,0,1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

VAR num vrefn:=100;
VAR num velmemoria:=100;
VAR intnum time_int;
VAR intnum time_int2;
VAR num override;
VAR num i:=0;
var num posnmemoria:=10;
var num veldata{1000};
var pos posdata{1000};
var num sensordata{1000};
var num tempodata{1000};
var num xforcex{1000};
var num yforcey{1000};
var num zforcez{1000};
var num txforcex{1000};
var num tyforcey{1000};
var num tzforcez{1000};
VAR Num posn:=-10;
VAR bool Flagpos :=false;
VAR loaddata discotool_LD:=[ 4.95, [0.764321, 0.718417, 139.852], [1, 0, 0, 0], 0, 0, 0];
VAR loaddata sapatotool_LD:=[ 4.28276, [-2.57186, 0.126338, 133.377], [1, 0, 0, 0], 0, 0, 0];
VAR fcforcevector myForceVector;
Var Num tempo;
Var clock clock2;
Var clock clock1;
PROC main()
```

```
ClkReset clock2;
ClkStart clock2;

FCDeact;
FCCalib sapatotool_LD;

pneusapato;
!sapatomachin;
!path_1;
!discomachin;

ClkStop clock2;
tempo := ClkRead (clock2);
TPWrite "tempo de execuçao" \Num:= tempo;

write_to_file;

ENDPROC

PROC sapatomachin()
      FCDeact;
      FCCalib sapatotool_LD;

      CONNECT time_int2 WITH write_data_toarray;
      ITimer 0.2, time_int2;
      ISleep time_int2;
      IWatch time_int2;

      ClkReset clock1;
      ClkStart clock1;


FCSpdChgAct 10\NonStopAllTime;
Path_4;
ClkStop clock1;
FCSpdChgDeact;
IDelete time_int2;
ENDPROC
```

```
PROC discomachin()
      FCDeact;
      FCCalib discotool_LD;
      Path_2;
ENDPROC
PROC Path_4()
      MoveJ p136,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
      MoveJ p137,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
      MoveJ p138,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
      MoveJ p97,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
      MoveJ p98,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
      MoveJ p99,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
      MoveJ p100,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
      MoveJ p101,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
      MoveJ p102,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
      MoveJ p103,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
      MoveJ p104,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
      MoveJ p105,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
      MoveJ p106,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
      MoveJ p107,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
      MoveJ p108,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
      MoveJ p109,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
      MoveJ p110,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
      MoveJ p111,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
      MoveJ p112,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
      MoveJ p113,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
      MoveJ p114,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
      MoveJ p115,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
      MoveJ p116,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
      MoveJ p117,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
      MoveJ p118,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
      MoveJ p119,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
      MoveJ p120,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
      MoveJ p121,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
      MoveJ p122,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
      MoveJ p123,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
      MoveJ p124,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
      MoveL p125,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
      MoveL p126,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
      MoveJ p127,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
```

```
        MoveJ p128,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p129,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p130,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p131,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p132,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p133,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p134,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p135,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p139,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p140,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p141,v100,z1,Tooldata_1\WObj:=Wobj_s_2;
ENDPROC

PROC pneusapato()
        CONNECT time_int2 WITH write_data_toarray;
        ITimer 0.2, time_int2;
        ISleep time_int2;
        IWatch time_int2;

        CONNECT time_int WITH speed_refresh;
        ITimer 0.1, time_int;
        ISleep time_int;
        override := CSpeedOverride (\CTask);
        IWatch time_int;

        MoveJ p136,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p137,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p138,v100,z10,Tooldata_1\WObj:=Wobj_s_2;

        MoveJ p97,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p98,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p99,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p100,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p101,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p102,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p103,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p104,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p105,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p106,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p107,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
```

```
        MoveJ p108,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p109,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p110,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p111,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p112,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p113,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p114,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p115,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p116,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p117,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p118,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p119,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p120,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p121,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p122,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p123,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p124,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveL p125,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveL p126,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p127,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p128,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p129,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p130,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p131,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p132,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p133,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p134,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p135,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p139,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p140,v100,z10,Tooldata_1\WObj:=Wobj_s_2;
        MoveJ p141,v100,z10,Tooldata_1\WObj:=Wobj_s_2;

        IDelete time_int;
        SpeedRefresh override;

        IDelete time_int2;

    ENDPROC

    PROC Path_2()
```

```
CONNECT time_int2 WITH write_data_toarray;
ITimer 0.2, time_int2;
ISleep time_int2;
IWatch time_int2;

MoveJ p26,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
MoveJ p31,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
MoveJ p30,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
MoveJ p29,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
MoveJ p25,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
      ClkReset clock1;
      ClkStart clock1;

FCSpdChgAct 10\NonStopAllTime;

MoveJ p24,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
MoveJ p23,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
MoveJ p22,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
MoveJ p21,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
MoveJ p20,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
MoveJ p19,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
MoveJ p18,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
MoveJ p17,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
MoveJ p16,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
MoveJ p15,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
MoveJ p14,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
MoveJ p13,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
MoveJ p12,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
MoveJ p11,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
MoveJ p10,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
MoveJ p9,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
MoveJ p8,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
MoveJ p7,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
MoveJ p6,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
MoveJ p5,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
MoveJ p4,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
MoveJ p3,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
MoveJ p2,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
MoveJ p1,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
MoveJ p23_2,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
```

```
        MoveJ p22_2,v100,z10,Tooldata_1\WObj:=Wobj_S_1;

        ClkStop clock1;
        FCSpdChgDeact;
        IDelete time_int2;

        MoveJ p26_3,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
ENDPROC

PROC Path_1()

        CONNECT time_int2 WITH write_data_toarray;
        ITimer 0.2, time_int2;
        ISleep time_int2;
        IWatch time_int2;

        CONNECT time_int WITH speed_refresh;
        ITimer 0.1, time_int;
        ISleep time_int;
        override := CSpeedOverride (\CTask);
        IWatch time_int;

        MoveJ p26,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
        MoveJ p31,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
        MoveJ p30,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
        MoveJ p29,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
        MoveJ p25,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
              ClkReset clock1;
              ClkStart clock1;

        vrefn:=100;
        MoveJ p24,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
        MoveJ p23,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
        MoveJ p22,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
        vrefn:=5;
        MoveJ p21,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
        MoveJ p20,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
        MoveJ p19,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
        vrefn:=100;
        MoveJ p18,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
```

```
        MoveJ p17,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
        MoveJ p16,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
        vrefn:=5;
        MoveJ p15,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
        MoveJ p14,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
        vrefn:=100;
        MoveJ p13,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
        MoveJ p12,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
        vrefn:=5;
        MoveJ p11,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
        MoveJ p10,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
        vrefn:=100;
        MoveJ p9,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
        vrefn:=5;
        MoveJ p8,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
        vrefn:=100;
        MoveJ p7,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
        vrefn:=5;
        MoveJ p6,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
        vrefn:=100;
        MoveJ p5,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
        vrefn:=5;
        MoveJ p4,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
        vrefn:=100;
        MoveJ p3,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
        vrefn:=5;
        MoveJ p2,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
        vrefn:=100;
        MoveJ p1,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
        MoveJ p23_2,v100,z10,Tooldata_1\WObj:=Wobj_S_1;
        MoveJ p22_2,v100,z10,Tooldata_1\WObj:=Wobj_S_1;

        ClkStop clock1;
        IDelete time_int;
        SpeedRefresh override;
        IDelete time_int2;

        MoveJ p26_3,v100,z10,Tooldata_1\WObj:=Wobj_S_1;

ENDPROC
```

```
TRAP speed_refresh

      GetData;
      SpeedRefresh vrefn;
      ERROR
      IF ERRNO = ERR_SPEED_REFRESH_LIM THEN
      IF vrefn > 100 vrefn := 100;
      IF vrefn < 0 vrefn := 0;
      RETRY;
      ENDIF
ENDTRAP

TRAP write_data_toarray

      i:=i+1;
      veldata{i}:= vrefn;
      sensordata{i}:= posn;
      posdata{i}:= Cpos (\Tool:=tool0\WObj:=wobj0);
      tempodata{i}:= ClkRead (clock2);
      myForceVector:= FCGetForce(\Tool:=Tooldata_1);

      xforcex{i}:= myForceVector.xforce;
      yforcey{i}:= myForceVector.yforce;
      zforcez{i}:= myForceVector.zforce;
      txforcex{i}:= myForceVector.xtorque;
      tyforcey{i}:= myForceVector.ytorque;
      tzforcez{i}:= myForceVector.ztorque;

ENDTRAP

PROC GetData()

      VAR string XDatast1:="";
      VAR string XDatast2:="";
      VAR Num XDatan1;
      VAR Num XDatan2;
      VAR byte XDataby1;
      VAR byte XDataby2;
      VAR byte XDatabyteste1;
```

```
VAR byte XDatabyteste2;
VAR byte XDatabyhi;
VAR byte XDatabylo;
VAR Num left_shift:=7;
VAR Num right_shift:=1;
VAR byte byHiInt;
VAR byte byHiInt2;
VAR byte ByHi;
VAR byte ByHi2;
VAR byte ByHi3;
VAR byte ByLO;
VAR Num NumCharacters:=1;
VAR bool bOK1;
VAR bool bOK2;
VAR bool Flagx :=false;
VAR iodev ComChannel;

      Open "COM1:", ComChannel \Bin;
      ClearIOBuff ComChannel;
      XDataby1:= ReadBin (ComChannel );
      XDataby2:= ReadBin (ComChannel );
      Close ComChannel;

      XDatabyteste1:= BitAnd(XDataby1,128);
      XDatabyteste2:= BitAnd(XDataby2,128);

      IF XDatabyteste1=128 THEN
            IF XDatabyteste2=0 THEN
            XDatabylo:=XDataby2;
            XDatabyhi:=XDataby1;
            Flagx:=true;
            endif

      elseIF XDatabyteste1=0 THEN
            IF XDatabyteste2=128 THEN
            XDatabyhi:=XDataby2;
            XDatabylo:=XDataby1;
            Flagx :=true;
            endif
      endIF
```

```
IF Flagx = true Then

ByHiInt:=BitAnd(XDatabyhi,1);
ByHiInt2:=BitLSh(ByhiInt,left_shift);
ByLo:=BitOr(ByHiInt2,XDatabylo);

ByHi:=BitAnd(XDatabyhi,126);
ByHi2:=BitRSh(ByHi,right_shift);
ByHi3:=BitAnd(ByHi2,63);

XDatast1:=ByteToStr(ByHi3);
XDatast2:=ByteToStr(ByLo);
bok1:=StrToVal(XDatast1,XDatan1);
bok2:=StrToVal(XDatast2,XDatan2);

posn:=XDatan1*256+XDatan2;
posnmemoria:=posn;
endif

IF Flagx = false Then
      TPWrite "erro de leitura de posicao";
      posn :=posnmemoria;

endif

If posn >= 35 then
      if Flagpos = false then
      velmemoria:= 10;
      Flagpos :=true;

      endif

elseif posn <30 then
      if Flagpos = true then
      Flagpos:=false;
      velmemoria:=100;

      endif
```

```
			endif
			vrefn:= velmemoria;

		!TPWrite "valor de vel no calculo " \Num:=vrefn;

ENDPROC

proc write_to_file()
		var iodev file;
		var num number :=8;
		var robtarget target;
		var pos posi1;
		var pos posi2;



		open "home:" \file :=  "file1.txt", file;
		write file,"tempo de execuçao programa:"\Num :=tempo;
		for i from 1 to 1000 DO
		write file,"leitura :"\Num :=i;
		write file, "ref vel  "\num :=veldata{i};
		write file, "pos data  "\pos :=posdata{i};
		write file, "sensor data  "\num :=sensordata{i};
		write file, "tempo  data  "\num :=tempodata{i};
		write file, "forcesensor  xdata  "\num :=xforcex{i};
		write file, "forcesensor  ydata  "\num :=yforcey{i};
		write file, "forcesensor  zdata  "\num :=zforcez{i};
		write file, "forcesensor  txdata  "\num :=txforcex{i};
		write file, "forcesensor  tydata  "\num :=tyforcey{i};
		write file, "forcesensor  tzdata  "\num :=tzforcez{i};

		ENDFOR
		close file;
ENDPROC

ENDMODULE
```